

A100780

AFIT/GE/EE/80D-13

14

6
DESIGN OF LINEAR PHASE,
FINITE IMPULSE RESPONSE,
TWO-DIMENSIONAL, DIGITAL FILTERS.

THESIS

9 Master thesis

Presented to the Faculty of the School of Engineering
of the Air Force Institute of Technology
Air University
in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

12 158

10 by
David Ciccolella, B.S.E.E
Capt USAF
Graduate Electrical Engineering

11 December 1980

Accession For	
NTIS GFHI	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability Codes	
Avail and/or Dist	Special
A	23
G	95

Approved for public release; distribution unlimited.

012225 LB

DISCLAIMER NOTICE

**THIS DOCUMENT IS BEST QUALITY
PRACTICABLE. THE COPY FURNISHED
TO DTIC CONTAINED A SIGNIFICANT
NUMBER OF PAGES WHICH DO NOT
REPRODUCE LEGIBLY.**

AFIT/GE/EE/80D-13

DESIGN OF LINEAR PHASE,
FINITE IMPULSE RESPONSE,
TWO-DIMENSIONAL, DIGITAL FILTERS
THESIS

AFIT/GE/EE/80D-13 David Ciccolella
 Capt USAF

Approved for public release; distribution unlimited.

Preface

I would like to thank all of the people who helped me with my research. I am especially indebted to Professor Larry Kizer for helping me acquire a background in digital signal processing and to Richard Brown for tutoring me on the McClellan Transformation and other aspects of the two-dimensional filter design problem. I am also indebted to my wife, Cecelia, who helped me solve some perplexing problems by providing a different point of view and fresh ideas.

David Ciccolella

Contents

Preface	ii
List of Figures and Tables	v
Abstract	vii
I. Introduction	1
Previous Development	2
Problem	4
General Approach	5
Sequence of Presentation	6
II. The McClellan Transformation and the Contour Approximation Problem	8
Generalized McClellan Transformation	8
Mechanics of the Transformation	10
Properties of the Contours	11
The Approximation Problem	13
Constraints	14
Linear Least-Squares Approximation with Constraints	16
Applicability of the Mapping	17
III. Mathematical Preliminaries for the Calculation of the Two-Dimensional Impulse Response	21
Chebyshev Polynomials of the First Kind	21
Forward Chebyshev Recursion: One Variable	23
Backward Chebyshev Recursion: One Variable	26
Backward Chebyshev Recursion: Two Variables	30
IV. Calculation of the Two-Dimensional Impulse Response	33
Form of the Two-Dimensional Digital Filters	33
One-Dimensional to Two-Dimensional Filter Transformation	35
Order of the Transformation	37
Summary	38

V.	Description of the Two-Dimensional Filter Design Program	39
	Overview	39
	Program CONTROL	44
	Program CURFIT	44
	Program PROTYPE	46
	Program FORCHEB	48
	Program EXPAND	48
	Program BACKCHB	48
	Program Graph	50
	Summary	50
		53
VI.	Design Results	55
	Filter Design 1	55
	Filter Design 2	58
	Filter Design 3	63
	Filter Design 4	63
	Filter Design 5	68
	Summary	72
VII.	Conclusions and Recommendations	75
	Conclusions	75
	Recommendations for Future Work	76
	Bibliography	78
	Appendix A: User's Guide for the Two-Dimensional Digital Filter Design Program	80
	Appendix B: Computer Listing of the Two-Dimensional Digital Filter Design Program	114
	Vita	146

List of Figures and Tables

Figure		Page
1	Contours Generated When $c(0,0) = .3422$, $c(1,0) = .5$, $c(0,1) = .5$, and $c(1,1) = -.3422$.	12
2	Typical Design Times for Two-Band Filters .	42
3	High Level Flowchart of the Design Program .	43
4	Flowchart of Program CONTROL . . .	45
5	High Level Flowchart of Program CURFIT . .	47
6	High Level Flowchart of Program FORCHEB . .	49
7	High Level Flowchart of Program BACKCHB . .	51
8	High Level Flowchart of Program GRAPH . .	54
9	Filter Design 1: Two-Dimensional Contours .	56
10	Filter Design 1: One-Dimensional Frequency Response	57
11	Filter Design 1: Two-Dimensional Frequency Response	59
12	Filter Design 2: Two-Dimensional Contours .	60
13	Filter Design 2: One-Dimensional Frequency Response	61
14	Filter Design 2: Two-Dimensional Frequency Response	62
15	Filter Design 3: Two-Dimensional Contours .	64
16	Filter Design 3: One-Dimensional Frequency Response	65
17	Filter Design 3: Two-Dimensional Frequency Response	66
18	Filter Design 4: Two-Dimensional Contours .	67
19	Filter Design 4: One-Dimensional Frequency Response	69
20	Filter Design 4: Two-Dimensional Frequency Response	70

Figure	Page
21 Filter Design 5: Two-Dimensional Contours	71
22 Filter Design 5: One-Dimensional Frequency Response	73
23 Filter Design 5: Two-Dimensional Frequency Response	74
24 Typical Three Band Filter	89
25 Design Example: Two-Dimensional Contours	110
26 Design Example: One-Dimensional Frequency Response	111
27 Design Example: Two-Dimensional Frequency Response	112

Table	Page
I. Chebyshev Polynomials of the First Kind	22
II. Powers of x in Terms of $T_n(x)$	23

Abstract

An interactive computer program was developed that enables the user to design linear phase, finite impulse response, linear shift-invariant, two-dimensional digital filters. The program user can design lowpass, highpass, bandpass, bandstop, all-pass, and multiband two-dimensional digital filters. The filters designed by using the program are nearly optimal in the Chebyshev sense and their magnitude versus frequency characteristics have quadrantal symmetry ($|H(w_2, w_1)| = |H(-w_2, w_1)| = |H(w_2, -w_1)| = |H(-w_2, -w_1)|$).

The technique implemented in the program consists of transforming a one-dimensional digital filter into a two-dimensional digital filter by a change of variables. This technique was first proposed by James H. McClellan and is called the McClellan Transformation. The program user can elect to utilize either the first order or the second order McClellan Transformation to design a two-dimensional digital filter.

DESIGN OF LINEAR PHASE,
FINITE IMPULSE RESPONSE,
TWO-DIMENSIONAL, DIGITAL FILTERS

I Introduction

The use of digital filters for signal processing applications is becoming pervasive. As the cost of digital components continues to decrease and the performance of the components continues to increase, digital signal processing becomes more and more attractive. Among the many applications of digital filters are seismic processing, picture processing, and speech processing.

Digital filters can be categorized using the length of the impulse response as either infinite duration impulse response filters (nonrecursive) or finite duration impulse response filters (recursive) (Ref 6:18). Initially, infinite duration impulse response (IIR) filters were more popular than finite duration impulse response (FIR) filters. FIR filters were generally felt to be inferior because long impulse response sequences were required in order to produce filters with sharp cutoff characteristics. However, with the development of the fast Fourier transformation (FFT) algorithm, implementation of high-order FIR filters can be made extremely computationally efficient (fast). FIR filters possess very desirable properties from the point of

view of filter design. First, FIR filters realized non-recursively are always stable (Ref 7:76). Second, FIR filter designs can always be realized with an appropriate finite time delay, and third FIR filters can be designed so that their frequency responses have exact linear phase characteristics.

Digital filters can also be categorized in terms of the dimension of the filter. For a one-dimensional filter, the frequency response is a function of one independent variable such as time. For a two-dimensional filter, the frequency response is a function of two independent variables. For example, picture processing is a two-dimensional filtering operation where two independent variables are the spacial coordinates of the picture.

Previous Development

Many techniques have recently been used to design two-dimensional FIR filters. Most of the techniques are extended versions of the analogous one-dimensional FIR digital filter design methods. Several of the more promising of these methods will be briefly described.

Two-dimensional FIR digital filters can be designed by multiplying the infinite duration ideal frequency response ($h(n_1, n_2)$) by a window function ($wd(n_1, n_2)$). Huang has explored the use of this technique (Ref 16). Basically Huang has shown that good two-dimensional windows can be obtained from good one-dimensional windows via the relation

$wd(n_1, n_2) = w1d((n_1^2 + n_2^2)^{\frac{1}{2}})$; where $w1d$ is an appropriate one-dimensional window function. Using this relation, the filter designer can design two-dimensional filters by using two-dimensional windows analogous to the well known one-dimensional windows (rectangular, Hamming, Kaiser, etc.). Windowing produces a frequency response which has ripples or overshoots (magnitude depends on the window used) at the discontinuities of piecewise constant ideal responses. This behavior is analogous to the Gibbs' effect in one dimension. Another problem with this method is that it is a convolution process in the frequency domain and thus discontinuities in the ideal response are smeared. Windowing has proven very useful because of its speed and its flexibility in approximating arbitrary ideal frequency responses.

Optimal two-dimensional FIR digital filters can be designed by using linear programming. Hu and Rabiner have explored the use of this approach (Ref 17). The linear programming method involves solving a set of linear inequalities in order to minimize the maximum error of the two-dimensional frequency response. Although linear programming is very flexible and can be used to approximate a wide variety of desired filter shapes, it is comparatively slow and thus its use is limited to the design of small order filters. The largest filter presented by Hu and Rabiner is a 9X9 sample points filter. The design involved solving several thousand constraint equations and took several hours on a high speed digital computer (IBM 370).

Two-dimensional FIR digital filters can also be designed by the transformation of variable technique. This method, proposed by McClellan, is several orders of magnitude faster than the windowing and linear programming methods for the design of a large class of two-dimensional filters (Ref 3:1-2). This technique is essentially a direct transformation of a one-dimensional filter design into a two-dimensional filter design. The transformation of variable method is very fast and can design large order two-dimensional filters in a matter of seconds on a general purpose digital computer. This technique will be described in detail in latter sections.

Problem

The goal of this investigation is to develop an interactive computer program to design a class of two-dimensional digital filters. In this context, the word "design" means the calculation of the impulse response coefficients necessary to approximate a desired frequency response. The structural implementation (direct, cascade, etc.) of the filter is not addressed in this investigation. All filters designed by the program will be linear phase, finite duration impulse response, and linear-shift invariant.

The technique to be used is the transformation of variable method first suggested by McClellan (Ref 3). This technique dictates that all filters have quadrantal symmetry.

This means that the two-dimensional magnitude versus frequency characteristic of any filter designed by using this technique will be symmetric in all four quadrants of the two-dimensional frequency plane. The program will allow the user the option of selecting either the first order or the second order McClellan Transformation for the two-dimensional filter design. The advantages and disadvantages of using the first or second order McClellan Transformation will be discussed in latter sections.

General Approach

This investigation will develop a computer program that implements the concepts proposed by McClellan (Ref 3) and Mersereau, Mecklenbrauker, and Quatieri (Ref 2) for the design of two-dimensional, linear phase, digital filters. The program will be an extension of their work in that it will provide the user with a systematic method of designing two-dimensional, linear phase, digital filters with a wide variety of shapes.

The algorithm used to develop the two-dimensional filter design program consists of the following major steps:

1. Define the shape of the magnitude versus frequency characteristic of the desired two-dimensional frequency response by specifying the shape and location of the contour in the w_2, w_1 plane that is to be mapped to the user's one-dimensional frequency. w_2 and w_1 are the variables on the axes of the plane in which the magnitude characteristic of the two-dimensional

frequency response is defined.

2. Perform a linear least-squares approximation with constraints in order to find the values of the constants of the McClellan Transformation that will produce the closest mapping of the user's one-dimensional frequency to the two-dimensional contour in the w_2, w_1 plane specified in step 1.
3. Design the one-dimensional prototype filter that will be transformed into the desired two-dimensional filter.
4. Calculate the impulse response coefficients of the resulting two-dimensional filter.

Sequence of Presentation

Chapter II starts with the McClellan Transformation and ends with the approximation problem. In the section on the McClellan Transformation, the mechanics of the transformation are discussed as are the properties of the contours produced by the transformation. The section on the approximation problem discusses the method of linear least-squares approximation with constraints.

Chapter III presents the tools and derivations necessary to understand the calculation of the two-dimensional impulse response and then chapter IV presents the calculation of the two-dimensional impulse response.

Chapter V presents the two-dimensional filter design program developed in this investigation. Chapter VI presents

design results obtained by using the two-dimensional filter design program. Chapter VII presents conclusions of this investigation as well as recommendations for future related work.

Two appendices conclude the thesis. Appendix A is a user's manual for the two-dimensional filter design program developed during this investigation and appendix B is a listing of the actual computer program.

II The McClellan Transformation and the Contour Approximation Problem

The design of two-dimensional digital filters as pursued in this investigation can be divided into two general areas. The first area concerns approximating the shape of the desired magnitude versus frequency characteristic of the two-dimensional filter by calculating the constants of the McClellan Transformation. The second area concerns the problem of calculating the impulse response of the resulting two-dimensional filter. The impulse response constitutes the desired filter design. In this chapter, the first of the two general areas will be explored.

Generalized McClellan Transformation

McClellan discovered that the equation describing the frequency response of a one-dimensional, FIR, linear phase filter

$$H(w) = e^{-jwk} \left(\sum_{n=0}^k h(n) \cos(wn) \right) \quad (1)$$

where

$H(w)$ = one-dimensional frequency response
 $h(n)$ = one-dimensional impulse response
 w = one-dimensional frequency variable

could be transformed into an equation describing the frequency response of a two-dimensional, FIR, linear phase filter by using a change of variables. The change of variables first proposed by McClellan is (Ref 3)

$$\cos(w) = \sum_{i=0}^1 \sum_{g=0}^1 \cos(iw_1) \cos(gw_2) c(i, g) \quad (2)$$

where the $c(i, g)$'s are the constants that determine the shape of the two-dimensional frequency response, and w_1 and w_2 are the two-dimensional frequency variables. This change of variables is called the McClellan Transformation.

Other researchers have generalized McClellan's original transformation by allowing the summation indices to assume values other than one. The generalized formulation of the McClellan transformation is (Ref 2)

$$m(w_2, w_1) = \sum_{i=0}^a \sum_{g=0}^b c(i, g) \cos(iw_1) \cos(gw_2) \quad (3)$$

For the first order McClellan Transformation ($a = b = 1$)

$$m(w_2, w_1) = c(0,0) + c(1,0) \cos(w_1) + c(0,1) \cos(w_2) \\ + c(1,1) \cos(w_1) \cos(w_2) \quad (4)$$

For the second order McClellan Transformation ($a = b = 2$)

$$m(w_2, w_1) = c(0,0) + c(1,0) \cos(w_1) + c(0,1) \cos(w_2) \\ + c(1,1) \cos(w_1) \cos(w_2) + c(1,2) \cos(w_1) \cos(2w_2) \\ + c(2,1) \cos(2w_1) \cos(w_2) + c(2,0) \cos(2w_1) \\ + c(0,2) \cos(2w_2) + c(2,2) \cos(2w_1) \cos(2w_2) \quad (5)$$

Using the McClellan Transformation constrains the magnitude versus frequency characteristic of the resulting two-

dimensional filter to have quadrantal symmetry ($|H(w_2, w_1)| = |H(-w_2, w_1)| = |H(w_2, -w_1)| = |H(-w_2, -w_1)|$). This means that the magnitude characteristic will be symmetric in all four quadrants of the two-dimensional frequency plane. This is true because the transformation equation possesses quadrantal symmetry and any symmetry that appears in the transformation equation also appears in the magnitude versus frequency characteristic of the resulting two-dimensional filter (Ref 1). This investigation explores the use of the first and second order versions of the generalized McClellan Transformation.

Mechanics of the Transformation. In order to help the reader understand the concepts that will be presented, only the form of the McClellan Transformation given by equation (4) will be treated. This will simplify the mathematics and allow the concepts to stand out. A parallel argument holds for equation (5).

When $m(w_2, w_1)$ in equation (4) is replaced by $\cos(w)$, equation (4) defines a mapping from the interval $[0, \pi]$ of the one-dimensional frequency axis to the square region $[0, \pi] \times [0, \pi]$ in the two-dimensional frequency plane (Ref 3). Making the substitution $\cos(w) = m(w_2, w_1)$ and solving equation (4) for w_1 as a function of w_2 yields

$$\cos(w_1) = \frac{\cos(w) - c(0,0) - c(0,1)\cos(w_2)}{c(1,0) + c(1,1)\cos(w_2)} \quad (6)$$

$$w_1 = \arccos \left[\frac{\cos(w) - c(0,0)\cos(w_2)}{c(1,0) + c(1,1)\cos(w_2)} \right] \quad (7)$$

From equation (7) it is easy to see that for a fixed w , there corresponds a curve in the w_2, w_1 plane. Along this curve, the transformed magnitude of the two-dimensional frequency response is a constant equal to the value of the magnitude of the one-dimensional frequency response at w . As w varies, a family of curves is generated that completely describes the transformed magnitude versus frequency characteristic of the two-dimensional frequency response. For example, if $c(0,0) = .3422$, $c(1,0) = .5$, $c(0,1) = .5$, and $c(1,1) = -.3422$ then the contours of figure 1 are generated.

Properties of the Contours. Part of the process of designing a two-dimensional filter involves choosing or calculating the $c(i,g)$'s in equation (4) or (5) in order to obtain a desired contour in the two-dimensional plane as one of the contours of constant w . Before considering how to perform this operation, it is necessary to discuss the allowable shapes of the contours in the two-dimensional frequency plane.

For the first order McClellan Transformation, the contours must be monotonic. This can be shown by letting $\cos(w)$ be fixed in equation (6) and taking the derivative of $\cos(w_1)$ with respect to $\cos(w_2)$.

$$\cos(w_1) = \frac{\cos(w) - c(0,0) - c(0,1)\cos(w_2)}{c(1,0) + c(1,1)\cos(w_2)} \quad (6)$$

$$\frac{d(\cos(w_1))}{d(\cos(w_2))} = \frac{-c(1,0)c(0,1) - c(1,1)\cos(w) + c(0,0)c(1,1)}{[c(1,0) + c(1,1)\cos(w_2)]^2} \quad (8)$$

W1-AXIS (X PI)

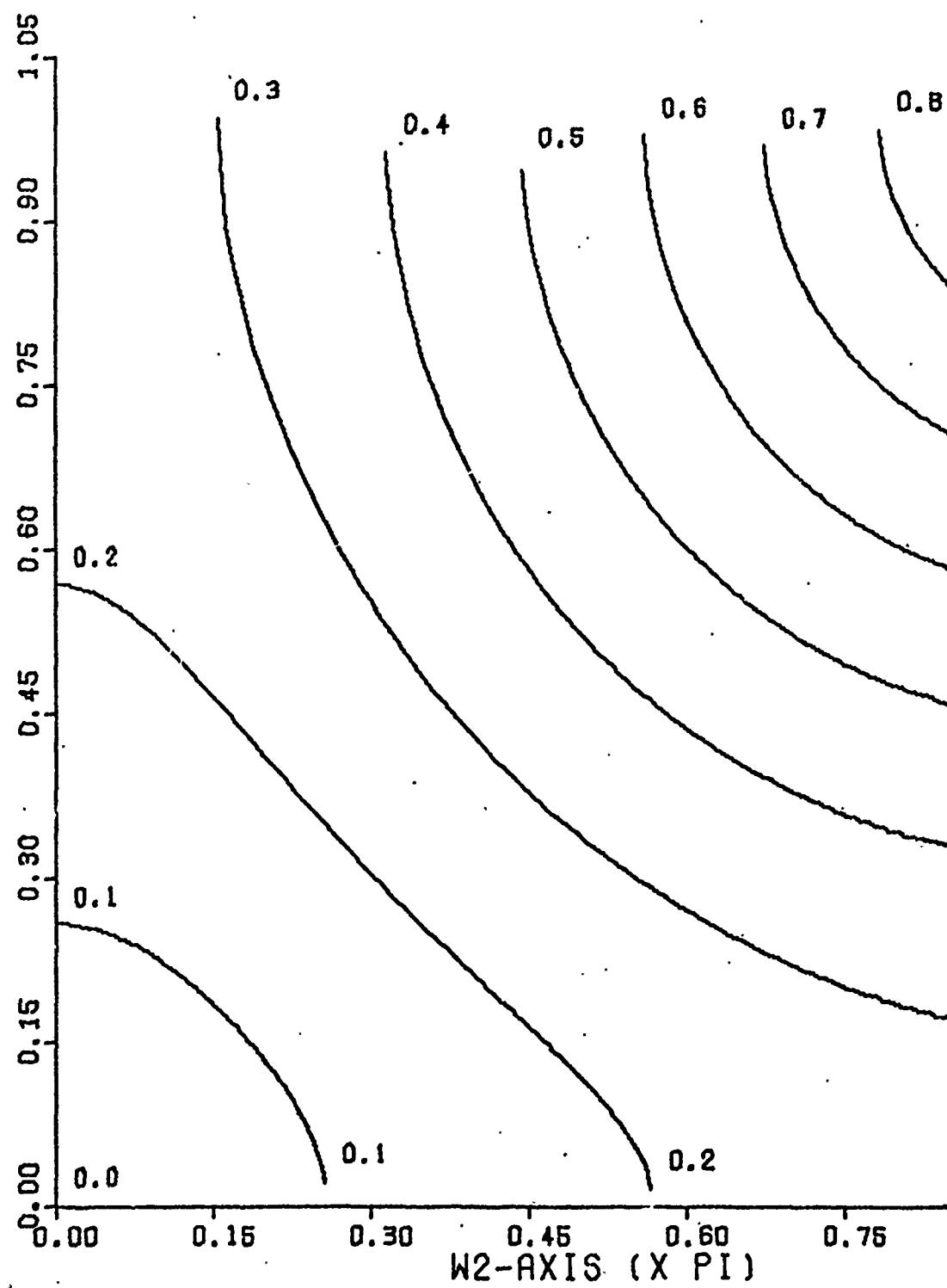


Fig 1. Contours Generated When $c(0,0) = .3422$, $c(1,0) = .5$,
 $c(0,1) = .5$, and $c(1,1) = -.3422$

Equation (8) shows that $\cos(w_1)$ is a monotonic function of $\cos(w_2)$ since the sign of the derivative does not change as $\cos(w_2)$ varies from -1 to +1. Therefore w_1 is a monotonic function of w_2 .

For the second order McClellan Transformation, it can be shown by a similar analysis that the requirement for monotonic contours does not apply (Ref 4:44-45). In other words, w_1 is not constrained to be a monotonic function of w_2 .

The Approximation Problem

This section is concerned with the problem of choosing the transformation parameters ($c(i,g)$'s of equation (3)) so that the resulting contours in the w_2, w_1 plane have some desired shape. In general only the shape of a single contour can be closely approximated using the method to be described. This is usually not a problem when designing the most common types of filters (lowpass, highpass, and all-pass) since the most important contour for approximation is the band edge. The shape of the other contours in the passband or stopband is usually not important. This constraint on the number of contours that can be closely approximated becomes more troublesome when designing bandpass, bandstop, and multi-band filters. Here the best that the designer can do is hope that all contours in the region of interest have the same shape. If this is true, the designer can usually get the band edges to lie where he desires by making adjustments to the $c(i,g)$'s of equation (3).

In the rest of this chapter, the second order McClellan Transformation will be used in the discussion. This will allow the reader to easily convert to the first order case by setting $c(0,2)$, $c(2,0)$, $c(2,2)$, $c(1,2)$, and $c(2,1)$ equal to zero in the equations that follow.

Constraints. The equation for the second order McClellan Transformation is

$$\begin{aligned}
 \cos(w) = & c(0,0) + c(1,0)\cos(w_1) + c(0,1)\cos(w_2) \\
 & + c(1,1)\cos(w_1)\cos(w_2) + c(1,2)\cos(w_1)\cos(2w_2) \\
 & + c(2,1)\cos(2w_1)\cos(w_2) + c(2,0)\cos(2w_1) \\
 & + c(0,2)\cos(2w_2) + c(2,2)\cos(2w_1)\cos(2w_2) \quad (9)
 \end{aligned}$$

For the approximation problem, w is fixed and is the one-dimensional frequency that is to map to the chosen contour in the w_2, w_1 plane. If the $c(i,g)$'s in equation (9) are allowed to assume any values then the trivial solution $c(0,0) = \cos(w)$, $c(1,0) = c(0,1) = c(1,1) = c(1,2) = c(2,1) = c(2,0) = c(0,2) = c(2,2) = 0.0$ will result in zero error. This solution results in the frequency w mapping to the entire two-dimensional plane $[0, \pi] \times [0, \pi]$. To prevent the trivial solution from occurring, some constraints must be placed on the $c(i,g)$'s.

The filter designer is free to choose any set of constraints so long as they do not conflict with the desired shape of the contour in the two-dimensional frequency plane.

It is to the advantage of the filter designer to choose a small set of constraints since each constraint forces one $c(i,g)$ to become a function of the remaining $c(i,g)$'s. Thus the approximation process is impaired if a large number of constraints is used and the desired contour has a complex shape.

To give the approximation process the most freedom, one constraint should be chosen. The following constraints are fairly independent of contour shape. If the constraint that $w = 0$ maps to $(w_2, w_1) = (0,0)$ is chosen, the constraint equation is

$$\begin{aligned} \cos(0) = & c(0,0) + c(1,0)\cos(0) + c(0,1)\cos(0) \\ & + c(1,1)\cos(0)\cos(0) + c(1,2)\cos(0)\cos(0) \\ & + c(2,1)\cos(0)\cos(0) + c(2,0)\cos(0) \\ & + c(0,2)\cos(0) + c(2,2)\cos(0)\cos(0) \end{aligned} \quad (10)$$

or

$$\begin{aligned} 1 = & c(0,0) + c(1,0) + c(0,1) + c(1,1) + c(1,2) \\ & + c(2,1) + c(2,0) + c(0,2) + c(2,2) \end{aligned} \quad (11)$$

If the constraint that $w = \pi$ maps to $(w_2, w_1) = (\pi, \pi)$ is chosen then the constraint equation is

$$\begin{aligned} -1 = & c(0,0) - c(1,0) - c(0,1) + c(1,1) - c(1,2) \\ & - c(2,1) + c(2,0) + c(0,2) + c(2,2) \end{aligned} \quad (12)$$

If the constraints that $w = 0$ maps to $(w_2, w_1) = (0,0)$ and $w = \pi$ maps to $(w_2, w_1) = (\pi, \pi)$ are chosen then the set of constraint equations consists of equations (11) and (12).

In summary, it has been shown that some constraints must be placed on the $c(i,g)$'s in order to achieve a useful solution. Several typical constraints were presented.

Linear Least-Squares Approximation With Constraints.

For the linear least squares approximation with constraints, w is fixed and represents a set of identical values. The basis functions used for the approximation are

$$g_1(w_2, w_1) = 1.0 \quad (13)$$

$$g_2(w_2, w_1) = \cos(w_1) \quad (14)$$

$$g_3(w_2, w_1) = \cos(w_2) \quad (15)$$

$$g_4(w_2, w_1) = \cos(w_1)\cos(w_2) \quad (16)$$

$$g_5(w_2, w_1) = \cos(w_1)\cos(2w_2) \quad (17)$$

$$g_6(w_2, w_1) = \cos(2w_1)\cos(w_2) \quad (18)$$

$$g_7(w_2, w_1) = \cos(2w_1) \quad (19)$$

$$g_8(w_2, w_1) = \cos(2w_2) \quad (20)$$

$$g_9(w_2, w_1) = \cos(2w_1)\cos(2w_2) \quad (21)$$

Only the first four basis functions are used when the first order McClellan Transformation is used in the approximation process. During the least squares approximation process

up to 18 (up to 8 when the first order transformation is used) simultaneous equations are solved. The solution produces the $c(i,g)$'s of equation (9). A detailed explanation of the linear least squares approximation with constraints method used in this investigation can be found in reference 15:95-100.

Applicability of the Mapping. For the transformation to be meaningful, the $c(i,g)$'s calculated from the least squares approximation process must be such that

$$-1 \leq \sum_{i=0}^2 \sum_{g=0}^2 c(i,g) \cos(iw_1) \cos(gw_2) \leq 1 \quad (22)$$

where $0 \leq w_1 \leq \pi$ and $0 \leq w_2 \leq \pi$. This is easily seen since in the transformation the expression of equation (22) is set equal to $\cos(w)$ (equation(9)). Values outside the allowable range of equation (22) correspond to complex values of w (Ref 2:408).

The filter designer can do three things if equation (22) is found not to hold after the linear least squares approximation has been performed. The first is to change the fixed value of w that was used the first time and then perform the approximation again. In the majority of cases this will not work. The reason is that the shape of the two-dimensional contours primarily determines the values of the $c(i,g)$'s. Even large changes in $\cos(w)$ only cause small changes in the $c(i,g)$'s. Also when equation (22) fails to hold, values much greater than +1 or much less than -1 are usually generated. These cannot be offset by changing $\cos(w)$ since

the range of $\cos(w)$ is small ($-1 \leq \cos(w) \leq +1$).

The second course of action is shifting and scaling. Shifting and scaling does not change the shape of the contours but it does change the value of the one-dimensional frequency associated with each contour (Ref 2:408). Shifting and scaling substitutes

$$c_1 \cos(w) - c_2 = c_1 \left[\sum_{i=0}^2 \sum_{g=0}^2 c(i,g) \cos(iw_1) \cos(gw_2) \right] - c_2 \quad (23)$$

for

$$\cos(w) = \sum_{i=0}^2 \sum_{g=0}^2 c(i,g) \cos(iw_1) \cos(gw_2) \quad (24)$$

Let under-barred quantities represent the shifted and scaled versions of the original quantities and let \bar{F}_{\max} denote the maximum value of the expression in equation (22) and \bar{F}_{\min} denote its minimum value. Then by choosing $c_1 = 2/(\bar{F}_{\max} - \bar{F}_{\min})$ and $c_2 = c_1 \bar{F}_{\max} - 1$, it can be shown (Ref 2:408) that the expression

$$-1 \leq c_1 \left[\sum_{i=0}^2 \sum_{g=0}^2 c(i,g) \cos(iw_1) \cos(gw_2) \right] - c_2 \leq 1 \quad (25)$$

must always be true. To calculate c_1 and c_2 , the equations

$$-1.0 = c_1 \bar{F}_{\min} - c_2 \quad (26)$$

$$+1.0 = c_1 \bar{F}_{\max} - c_2 \quad (27)$$

are solved simultaneously for c_1 and c_2 . The solutions of equations (26) and (27) are

$$c_1 = 2/(\bar{F}_{\max} - \bar{F}_{\min}) \quad (28)$$

$$c_2 = c_1 F_{\max} - 1 \quad (29)$$

After shifting and scaling equation (24) can be rewritten as

$$\cos(\underline{w}) = \sum_{i=0}^2 \sum_{g=0}^2 \underline{c}(i,g) \cos(iw_1) \cos(gw_2) \quad (30)$$

where

$$\underline{w} = \arccos(c_1 \cos(w) - c_2) \quad (31)$$

$$\underline{c}(0,0) = c_1 c(0,0) - c_2 \quad (32)$$

$$\underline{c}(1,0) = c_1 c(1,0) \quad (33)$$

$$\underline{c}(0,1) = c_1 c(0,1) \quad (34)$$

$$\underline{c}(1,1) = c_1 c(1,1) \quad (35)$$

$$\underline{c}(1,2) = c_1 c(1,2) \quad (36)$$

$$\underline{c}(2,1) = c_1 c(2,1) \quad (37)$$

$$\underline{c}(2,0) = c_1 c(2,0) \quad (38)$$

$$\underline{c}(0,2) = c_1 c(0,2) \quad (39)$$

$$\underline{c}(2,2) = c_1 c(2,2) \quad (40)$$

Although shifting and scaling always produces a well-defined mapping from the w -axis to the w_2, w_1 plane, the result may or may not be satisfactory from a filter design point of view. Shifting and scaling changes the location of the original specified contour in the two-dimensional frequency plane. The new location may or may not be close to the original location.

The third course of action is to perform the least squares approximation with a different set of constraints. Changing the constraints has little or no effect on the shape of the contours produced by the approximation process

and it does not require the filter designer to change the desired values of parameters (such as w). For these reasons, this course of action is superior to the others when it works.

III Mathematical Preliminaries for the Calculation of the Two- Dimensional Impulse Response

In this chapter, the reader will be acquainted with the mathematical tools and concepts necessary in order to understand the presentation in chapter IV. The Chebyshev polynomials will be presented and several important relationships used in this investigation will be derived.

The Chebyshev polynomials are a very important mathematical tool, because they greatly simplify the analysis of the one-dimensional filter transformation process. With the aid of the Chebyshev polynomials, the analysis can be done in terms of polynomials rather than in terms of trigonometric functions.

Chebyshev Polynomials of the First Kind

Chebyshev polynomials of the first kind are polynomials in x^n (Ref 5:54-59). Their symbol is $T_n(x)$. The Chebyshev polynomials are defined by the relationship

$$T_n(x) = \cos(n \cos^{-1} x) \quad (41)$$

where n can be any integer (positive, negative, or zero) and x ranges from +1.0 to -1.0. If $n = 0$, then

$$T_0(x) = x^0 = 1.0 \quad (42)$$

If $n = 1$, then

$$T_1(x) = x^1 \quad (43)$$

In general, the following recursion relation allows the calculation of any Chebyshev polynomial if the preceding two are known:

$$T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x) \quad (44)$$

Just as the Chebyshev polynomials $T_n(x)$ can be written as polynomials in terms of the powers of x , the process can be reversed and the powers of x can be written in terms of $T_n(x)$. Table I gives the first 12 Chebyshev polynomials expressed in terms of the powers of x .

TABLE I
Chebyshev Polynomials of the First Kind

$$T_0(x) = 1$$

$$T_1(x) = x$$

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

$$T_5(x) = 16x^5 - 20x^3 + 5x$$

$$T_6(x) = 32x^6 - 48x^4 + 18x^2 - 1$$

$$T_7(x) = 64x^7 - 112x^5 + 56x^3 - 7x$$

$$T_8(x) = 128x^8 - 256x^6 + 160x^4 - 32x^2 + 1$$

$$T_9(x) = 256x^9 - 576x^7 + 432x^5 - 120x^3 + 9x$$

$$T_{10}(x) = 512x^{10} - 1280x^8 + 1120x^6 - 400x^4 + 50x^2 - 1$$

$$T_{11}(x) = 1024x^{11} - 2816x^9 + 2816x^7 - 1232x^5 + 220x^3 - 11x$$

Table II gives the first 12 powers of x expressed in terms of $T_n(x)$.

TABLE II
Powers of x in Terms of $T_n(x)$

$$x^0 = T_0$$

$$x^1 = T_1$$

$$x^2 = 1/2(T_2 + T_0)$$

$$x^3 = 1/4(T_3 + 3T_1)$$

$$x^4 = 1/8(T_4 + 4T_2 + 3T_0)$$

$$x^5 = 1/16(T_5 + 5T_3 + 10T_1)$$

$$x^6 = 1/32(T_6 + 6T_4 + 15T_2 + 10T_0)$$

$$x^7 = 1/64(T_7 + 7T_5 + 21T_3 + 35T_1)$$

$$x^8 = 1/128(T_8 + 8T_6 + 28T_4 + 56T_2 + 35T_0)$$

$$x^9 = 1/256(T_9 + 9T_7 + 36T_5 + 84T_3 + 126T_1)$$

$$x^{10} = 1/512(T_{10} + 10T_8 + 45T_6 + 120T_4 + 210T_2 + 126T_0)$$

$$x^{11} = 1/1024(T_{11} + 11T_9 + 55T_7 + 165T_5 + 330T_3 + 462T_1)$$

Forward Chebyshev Recursion: One Variable

Given the equation

$$\sum_{j=0}^k b(j)x^j = \sum_{j=0}^k a(j)T_j(x) \quad (45)$$

the forward Chebyshev recursion problem consists of calculating the $b(j)$'s when the $a(j)$'s are known. The equation merely states that the sum of a linear combination of Chebyshev polynomials can also be expressed as a polynomial in terms of x . A recursive relationship for the $b(j)$'s in

terms of the $a(j)$'s can be found by first expanding the right side of equation (45) (the x of $T_n(x)$ will be dropped for convenience).

$$\sum_{j=0}^k b(j)x^j = a(0)T_0 + a(1)T_1 + a(2)T_2 + a(3)T_3 + a(4)T_4 + \dots + a(k)T_k \quad (46)$$

Now using the relationships of equations (42), (43), and (44) or table I; the Chebyshev polynomials on the right side of equation (46) are written in terms of the powers of x .

$$\sum_{j=0}^k b(j)x^j = a(0)x^0 + a(1)x^1 + a(2)[2x^2 - 1] + a(3)[4x^3 - 3x] + a(4)[8x^4 - 8x^2 + 1] + \dots + a(k)[f(x)] \quad (47)$$

Now collecting like terms and ordering the powers of x on the right side of equation (47) yields

$$\begin{aligned} \sum_{j=0}^k b(j)x^j &= [a(0) - a(2) + a(4) + \dots]x^0 + [a(1) - 3a(3) + \dots]x^1 \\ &\quad + [2a(2) - 8a(4) + \dots]x^2 + [4a(3) + \dots]x^3 \\ &\quad + [8a(4) + \dots]x^4 + \dots + [\dots + \dots + \dots]x^k \end{aligned} \quad (48)$$

Expanding the left side of equation (48) and equating coefficients of like powers of x yields

$$b(0) = 2^0[a(0) - a(2) + a(4) - a(6) + a(8) - a(10) + \dots] \quad (49)$$

$$b(1) = 2^0[a(1) - 3a(3) + 5a(5) - 7a(7) + 9a(9) - 11a(11) + \dots] \quad (50)$$

$$b(2) = 2^1[a(2) - 4a(4) + 9a(6) - 16a(8) + 25a(10) - \dots] \quad (51)$$

$$b(3) = 2^2[a(3) - 5a(5) + 14a(7) - 30a(9) + 55a(11) - \dots] \quad (52)$$

$$b(4) = 2^3[a(4) - 6a(6) + 20a(8) - 50a(10) + \dots] \quad (53)$$

$$b(5) = 2^4 [a(5) - 7a(7) + 27a(9) - 77a(11) + \dots] \quad (54)$$

$$b(6) = 2^5 [a(6) - 8a(8) + 35a(10) - \dots] \quad (55)$$

$$\mathfrak{b}(7) = 2^6 [a(7) - 9a(9) + 44a(11) - \dots] \quad (56)$$

$$b(8) = 2^7 [a(8) - 10a(10) + \dots] \quad (57)$$

$$b(9) = 2^8 [a(9) - 11a(11) + \dots] \quad (58)$$

$$b(10) = 2^9 [a(10) + \dots] \quad (59)$$

$$b(11) = 2^{10} [a(11) + \dots] \quad (60)$$

$$b(k) = 2^{k-1} a(k) \quad (61)$$

Now putting equations (49) through (61) in matrix form yields

Call the large matrix D. The elements of D are related by the following formulas:

$$|D(0, m)| = \begin{cases} 1 & \text{for } m \text{ even} \\ 0 & \text{for } m \text{ odd} \end{cases} \quad (63)$$

$$|D(1, m)| = \begin{cases} 0 & \text{for } m \text{ even} \\ m & \text{for } m \text{ odd} \end{cases} \quad (64)$$

$$D(l, m) = 0 \text{ for } m < l \quad (65)$$

$$D(l, m) = 1 \text{ for } l = m \quad (66)$$

$$|D(l, m)| = |D(l, m-2)| + |D(l-1, m-1)| \text{ for } m > 1 \text{ and} \\ l \neq 0 \text{ and} \\ l \neq 1 \quad (67)$$

The signs of the non-zero elements of each row alternate between +1 and -1 starting at the main diagonal element in each row.

Thus it has been shown that the $b(j)$'s of equation (45) can always be calculated by constructing the matrix D and then applying equation (62).

Backward Chebyshev Recursion: One Variable

Given the equation

$$\sum_{j=0}^k a(j)T_j(x) = \sum_{j=0}^k b(j)x^j \quad (68)$$

the backward Chebyshev recursion problem consists of calculating the $a(j)$'s when the $b(j)$'s are known. Again, the equation merely states that the sum of a linear combination of Chebyshev polynomials can also be expressed as a polynomial in terms of x . A recursive relationship for the $a(j)$'s in

terms of the $b(j)$'s can be found by first expanding the right side of equation (68) (the x of $T_n(x)$ will again be dropped for convenience).

$$\sum_{j=0}^k a(j)T_j = b(0)x^0 + b(1)x^1 + b(2)x^2 + b(3)x^3 + b(4)x^4 + \dots + b(k)x^k \quad (69)$$

Now using the relationships of equations (42), (43), and (44) or table II; the powers of x on the right side of equation (69) are written in terms of the Chebyshev polynomials.

$$\sum_{j=0}^k a(j)T_j = b(0)T_0 + b(1)T_1 + b(2)[1/2(T_2+T_0)] + b(3)[1/4(T_3+3T_1)] + b(4)[1/8(T_4+4T_2+3T_0)] + \dots + b(k)[f(T(x))] \quad (70)$$

Now collecting like terms and ordering the Chebyshev polynomials on the right side of equation (70) yields

$$\sum_{j=0}^k a(j)T_j = [b(0)+1/2b(2)+3/8b(4)+\dots]T_0 + [b(1)+3/4b(3)+\dots]T_1 + [1/2b(2)+4/8b(4)+\dots]T_2 + [1/4b(3)+\dots]T_3 + [1/8b(4)+\dots]T_4 + \dots + [\dots+\dots+\dots]T_k \quad (71)$$

Expanding the left side of equation (71) and equating coefficients of like Chebyshev polynomials yields

$$a(0) = b(0) + 1/2b(2) + 3/8b(4) + 10/32b(6) + 35/128b(8) + 126/512b(10) + \dots \quad (72)$$

$$a(1) = b(1) + 3/4b(3) + 10/16b(5) + 35/64b(7) + 126/256b(9) + 462/1024b(11) + \dots \quad (73)$$

$$a(2) = 1/2b(2) + 4/8b(4) + 15/32b(6) + 56/128b(8) + 210/512b(10) + \dots \quad (74)$$

$$a(3) = 1/4b(3) + 5/16b(5) + 21/64b(7) + 84/256b(9) + 530/1024b(11) + \dots \quad (75)$$

$$a(4) = 1/8b(4) + 6/32b(6) + 28/128b(8) + 120/512b(10) + \dots \quad (76)$$

$$a(5) = 1/16b(5) + 7/64b(7) + 36/256b(9) + 165/1024b(11) + \dots \quad (77)$$

$$a(6) = 1/32b(6) + 8/128b(8) + 45/512b(10) + \dots \quad (78)$$

$$a(7) = 1/64b(7) + 9/256b(9) + 55/1024b(11) + \dots \quad (79)$$

$$a(8) = 1/128b(8) + 10/512b(10) + \dots \quad (80)$$

$$a(9) = 1/256b(9) + 11/1024b(11) + \dots \quad (81)$$

$$a(10) = 1/512b(10) + \dots \quad (82)$$

$$a(11) = 1/1024b(11) + \dots \quad (83)$$

$$\vdots$$

$$a(k) = 2^{-(k-1)}b(k) \quad (84)$$

Now putting equations (72) through (84) in matrix form yields

$$\begin{array}{c|c|c|c}
 a(0) & 1 & 0 & 1 & 0 & 3 & 0 & 10 & 0 & 35 & 0 & 126 & 0 & \dots & b(0) \\
 a(1) & 0 & 1 & 0 & 3 & 0 & 10 & 0 & 35 & 0 & 126 & 0 & 462 & \dots & b(1) \\
 a(2) & 0 & 0 & 1 & 0 & 4 & 0 & 15 & 0 & 56 & 0 & 210 & 0 & \dots & b(2) \\
 a(3) & 0 & 0 & 0 & 1 & 0 & 5 & 0 & 21 & 0 & 84 & 0 & 330 & \dots & b(3) \\
 a(4) & 0 & 0 & 0 & 0 & 1 & 0 & 6 & 0 & 28 & 0 & 120 & 0 & \dots & b(4) \\
 a(5) & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 7 & 0 & 36 & 0 & 165 & \dots & b(5) \\
 a(6) & = & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 8 & 0 & 45 & 0 & \dots & b(6) \\
 a(7) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 9 & 0 & 55 & \dots & b(7) \\
 a(8) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 10 & 0 & \dots & b(8) \\
 a(9) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 11 & \dots & b(9) \\
 a(10) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & \dots & b(10) \\
 a(11) & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \dots & b(11) \\
 \vdots & \dots & \vdots \\
 a(k) & \vdots & \dots & b(k) \\
 \hline
 & \overline{1} & \overline{1} & \overline{2} & \overline{4} & \overline{8} & \overline{16} & \overline{32} & \overline{64} & \overline{2^7} & \overline{2^8} & \overline{2^9} & \overline{2^{10}} & \dots & \overline{2^{k-1}}
 \end{array} \tag{85}$$

where the numbers at the bottom of each column indicate that all the entries in the column are to be divided by that number.

Call the large matrix C . The elements of C are related by the following formulas:

$$C(l, m) = 1 \text{ for } l = m \tag{86}$$

$$C(l, m) = 0 \text{ for } m < l \tag{87}$$

$$C(0, m) = C(1, m-1) \quad (88)$$

$$C(1, m) = 2C(0, m-1) + C(2, m-1) \quad (89)$$

$$C(l, m) = C(l-1, m-1) + C(l+1, m-1) \text{ for } m > l \text{ and } l \neq 1 \\ \text{and } l \neq 0 \quad (90)$$

Thus it has been shown that the $a(j)$'s of equation (68) can always be calculated by constructing the matrix C and then applying equation (85).

Backward Chebyshev Recursion: Two Variables

Given the equation

$$\sum_{i=0}^k \sum_{j=0}^k a(i, j) T_i(x) T_j(y) = \sum_{i=0}^k \sum_{j=0}^k b(i, j) x^i y^j \quad (91)$$

the backward Chebyshev recursion problem in two variables consists of converting the $b(i, j)$'s into the $a(i, j)$'s when the $b(i, j)$'s are known. This can be done by fixing i (denoted by i) and separating the summations on the right side of equation (91).

$$\sum_{i=0}^k \sum_{j=0}^k a(i, j) T_i(x) T_j(y) = \sum_{i=0}^k x^i \left[\sum_{j=0}^k b(i, j) y^j \right] \quad (92)$$

Now expanding the right side of equation (92) yields

$$\sum_{i=0}^k \sum_{j=0}^k a(i, j) T_i(x) T_j(y) = x^0 \sum_{j=0}^k b(0, j) y^j \\ + x^1 \sum_{j=0}^k b(1, j) y^j + \dots + x^k \sum_{j=0}^k b(k, j) y^j \quad (93)$$

Using the backward Chebyshev recursion in one variable, each individual summation on the right side of equation (93) can be converted from the form $\sum_{j=0}^k b(i, j) y^j$ to the form

$\sum_{j=0}^k d(i, j) T_j(y)$ by use of equation (85).

$$\sum_{i=0}^k \sum_{j=0}^k a(i, j) T_i(x) T_j(y) = x^0 \sum_{j=0}^k d(0, j) T_j(y) + x^1 \sum_{j=0}^k d(1, j) T_j(y) + \dots + x^k \sum_{j=0}^k d(k, j) T_j(y) \quad (94)$$

$$\sum_{i=0}^k x^i \left[\sum_{j=0}^k d(i, j) T_j(y) \right] \quad (95)$$

$$= \sum_{i=0}^k \sum_{j=0}^k d(i, j) x^i T_j(y) \quad (96)$$

Now fixing j (denoted by j) and separating the summation on the right side of equation (96) yields

$$\sum_{i=0}^k \sum_{j=0}^k a(i, j) T_i(x) T_j(y) = \sum_{j=0}^k T_j(y) \left[\sum_{i=0}^k d(i, j) x^i \right] \quad (97)$$

Expanding the right side of equation (97) yields

$$\sum_{i=0}^k \sum_{j=0}^k a(i, j) T_i(x) T_j(y) = T_0(y) \sum_{i=0}^k d(i, 0) x^i + T_1(y) \sum_{i=0}^k d(i, 1) x^i + \dots + T_k(y) \sum_{i=0}^k d(i, k) x^i \quad (98)$$

Using the backward Chebyshev recursion in one variable, each individual summation on the right side of equation (98) can be inverted from the form $\sum_{i=0}^k d(i, j) x^i$ to the form $\sum_{j=0}^k a(i, j) T_j(x)$ by use of equation (85).

$$\sum_{i=0}^k \sum_{j=0}^k a(i, j) T_i(x) T_j(y) = T_0(y) \sum_{i=0}^k a(i, 0) T_i(x) + T_1(y) \sum_{i=0}^k a(i, 1) T_i(x) + \dots + T_k(y) \sum_{i=0}^k a(i, k) T_i(x) \quad (99)$$

$$= \sum_{j=0}^k T_j(y) \left[\sum_{i=0}^k a(i,j) T_i(x) \right] \quad (100)$$

$$= \sum_{i=0}^k \sum_{j=0}^k a(i,j) T_i(x) T_j(y) \quad (101)$$

Thus it has been shown that the $a(i,j)$'s of equation (91) can always be calculated by performing the backward Chebyshev recursion in one variable twice.

This chapter has presented the Chebyshev polynomials and several derivations using these polynomials. The relationships developed in this chapter will greatly simplify the analysis of the transformation process that is presented in the next chapter.

IV Calculation of the Two-Dimensional Impulse Response

The theory describing the conversion of a one-dimensional digital filter into a two-dimensional digital filter will now be presented. In the first part of this chapter, the frequency response equation of one class of two-dimensional digital filters will be derived. The second part of the chapter will start with the general equation for the frequency response of a one-dimensional digital filter. Then by applying some constraints and using the McClellan Transformation, it will be shown that this equation can be transformed into the equation derived in the first part of the chapter. The chapter will conclude with a discussion of the advantages and disadvantages of using a particular order (first or second) transformation in the filter design process.

Form of the Two-Dimensional Digital Filters

In this section, constraints will be placed on the general equation describing the frequency response of two-dimensional digital filters. After the constraints have been applied, the resulting equation will describe the frequency response of the class of two-dimensional digital filters that can be designed by the transformation of variable process.

Let $H2D(l, m)$ be the real, causal, linear shift-

invariant, finite-duration, impulse response of a two-dimensional digital filter defined over the interval $0 \leq l \leq N_1-1$, $0 \leq m \leq N_1-1$. Taking the two-dimensional Fourier transformation of $H_{2D}(l, m)$ yields the two-dimensional frequency response, $H(w_2, w_1)$.

$$H(w_2, w_1) = \sum_{l=0}^{N_1-1} \sum_{m=0}^{N_1-1} H_{2D}(l, m) e^{-jw_1 l} e^{-jw_2 m} \quad (102)$$

The frequency response is a periodic function of w_2 and w_1 with a period of 2π . It can be shown (Ref 3:271) that if N_1 is constrained to be odd and the symmetry conditions

$$H_{2D}(N_1-1-d, m) = H_{2D}(d, m); \quad d = 0, 1, \dots, (N_1-1)/2 = k_1 \quad (103)$$

$$H_{2D}(l, N_1-1-d) = H_{2D}(l, d); \quad d = 0, 1, \dots, (N_1-1)/2 = k_1 \quad (104)$$

are imposed, then the two-dimensional frequency response can be written as

$$H(w_2, w_1) = e^{-jk_1(w_2+w_1)} \left[\sum_{l=0}^{k_1} \sum_{m=0}^{k_1} \underline{H_{2D}(l, m)} \cos(w_1 l) \cdot \cos(w_2 m) \right] \quad (105)$$

where (Ref 3:271)

$$\underline{H_{2D}}(0, 0) = H_{2D}(k_1, k_1) \quad (106)$$

$$\underline{H_{2D}}(0, m) = 2 \cdot H_{2D}(k_1, k_1-m); \quad m = 1, 2, \dots, k_1 \quad (107)$$

$$\underline{H_{2D}}(l, 0) = 2 \cdot H_{2D}(k_1-l, k_1); \quad l = 1, 2, \dots, k_1 \quad (108)$$

$$\underline{H_{2D}}(l, m) = 4 \cdot H_{2D}(k_1-l, k_1-m); \quad l = 1, \dots, k_1 \text{ and } m = 1, \dots, k_1 \quad (109)$$

One-Dimensional to Two-Dimensional Filter Transformation

In this section, the equation describing the frequency response of a class of one-dimensional filters will be transformed into the equation describing the frequency response of a class of two-dimensional digital filters. This will be done by first applying constraints to the equation describing the one-dimensional digital filters and then applying the McClellan Transformation.

Let $h(n)$ be the real, causal, linear shift-invariant, finite-duration, impulse response of a one-dimensional digital filter defined over the interval $0 \leq n \leq N_2-1$. Taking the Fourier transformation of $h(n)$ yields the frequency response, $H(w)$ (Ref 6:19-21).

$$H(w) = \sum_{n=0}^{N_2-1} h(n)e^{-jwn} \quad (110)$$

The frequency response is periodic in frequency with period 2π . It can be shown (Ref 7:77) that for one class of linear phase, FIR, digital filters

$$h(n) = h(N_2-1-n) \quad (111)$$

Using equation (111) and constraining N_2 to be odd, the frequency response can be written as (Ref 7:81-82)

$$H(w) = e^{-jwk_2} \sum_{n=0}^{k_2} H_1D(n) \cos(wn) \quad (112)$$

where

$$k_2 = (N_2-1)/2 \quad (113)$$

$$H_1D(0) = h(k_2) \quad (114)$$

$$H1D(n) = 2 \cdot h(k2-n); n = 1, 2, \dots, k2 \quad (115)$$

By letting $x = \cos(w)$ and using the definition of the Chebyshev polynomial (equation (41)), equation (112) becomes

$$H(w) = e^{-jwk2} \sum_{n=0}^{k2} H1D(n) T_n(x) \quad (116)$$

Using the forward Chebyshev recursion in one variable (equation (62)), equation (116) is converted to

$$H(w) = e^{-jwk2} \sum_{n=0}^{k2} H1DCHEB(n) x^n \quad (117)$$

Since $x = \cos(w)$, equation (117) can be rewritten as

$$H(w) = e^{-jwk2} \left[\sum_{n=0}^{k2} H1DCHEB(n) \cdot [\cos(w)]^n \right] \quad (118)$$

Now by substituting the generalized McClellan Transformation for $\cos(w)$ (equation (3))

$$H(w) = e^{-jwk2} \left[\sum_{n=0}^{k2} H1DCHEB(n) \cdot \left\{ \sum_{l=0}^a \sum_{m=0}^b c(l, m) \cos(lw_1) \cos(mw_2) \right\}^n \right] \quad (119)$$

In the generalized McClellan Transformation let $a = b$ and let the generalized McClellan Transformation be represented by P . Equation (119) can be partially expanded to yield

$$H(w) = e^{-jwk2} \left[H1DCHEB(0)P^0 + H1DCHEB(1)P^1 + \dots + H1DCHEB(k2)P^{k2} \right] \quad (120)$$

If equation (120) is completely expanded, the resulting terms can be regrouped by powers of $\cos(w_1)$ and $\cos(w_2)$ and

represented by the following summation (Ref 4:111-116)

$$H(w_2, w_1) = e^{-jwk_2} \left[\sum_{l=0}^{a \cdot k^2} \sum_{m=0}^{a \cdot k^2} H_{2D\text{CHEB}}(l, m) (\cos(w_1))^l \cdot (\cos(w_2))^m \right] \quad (121)$$

Now using the Chebyshev recursion in two variables (equations (91) through (101)), equation (121) is converted to

$$H(w_2, w_1) = e^{-jwk_2} \sum_{l=0}^{a \cdot k^2} \sum_{m=0}^{a \cdot k^2} H_{2D}(l, m) \cos(w_1 l) \cos(w_2 m) \quad (122)$$

Equation (122) is the desired result. It has the same form as equation (105).

Order of the Transformation

From the summation indices of equation (121), it can be seen that for any two-dimensional digital filter designed by the generalized McClellan Transformation (equation (3)); there are $(a \cdot k^2 + 1)(a \cdot k^2 + 1)$ or $[a \cdot (N_2 - 1)/2 + 1][a \cdot (N_2 - 1)/2 + 1]$ unique two-dimensional impulse response samples. For the first order transformation $a = 1$. The number of unique impulse response samples is $[(N_2 - 1)/2 + 1][(N_2 - 1)/2 + 1]$ or approximately $N_2^2/4$. For the second order transformation $a = 2$. The number of unique impulse response samples is $(N_2 - 1 + 1)(N_2 - 1 + 1) = N_2^2$. Therefore, for a given one-dimensional filter order, N_2 , the design using a second order transformation will produce approximately four times as many unique two-dimensional impulse response samples as the design using the first order McClellan Transformation. If the filter designer plans to implement the two-dimensional

filter design, it is obviously easier to implement a filter designed by using the first order McClellan Transformation.

Summary

In this chapter it has been demonstrated that a class of one-dimensional filters can be transformed into a class of two-dimensional filters via the McClellan Transformation. It has also been shown that for a given one-dimensional filter order, a design using the second order transformation generates approximately four times as many unique impulse response samples as a design using the first order transformation.

V Description of the Two-Dimensional Filter Design Program

The salient characteristics of the two-dimensional digital filter design program developed in this investigation are discussed in order to acquaint the reader with the program. The filter design program consists of seven separate programs. They are called CONTROL, CURFIT, PROTOTYPE, FORCHEB, EXPAND, BACKCHB, and GRAPH. After an overview of the program as a whole, each of the seven separate programs will be described.

Overview

The main objectives at the beginning of this investigation were to develop an interactive computer program that would design a class of two-dimensional digital filters and would be easy to use. The minimum output was to be the two-dimensional impulse response necessary to approximate a desired frequency response.

The two-dimensional digital filter design program developed in this investigation will design linear phase, finite impulse response, linear shift-invariant, two-dimensional, digital filters. This is done by transforming a one-dimensional digital filter with a frequency response of the form (chapter IV)

$$H(w) = e^{-jw \cdot k^2} \sum_{n=0}^{k^2} H_1 D(n) \cos(wn) \quad (112)$$

into a two-dimensional digital filter with a frequency response of the form (chapter IV)

$$H(w_2, w_1) = e^{-j \cdot k_1(w_2 + w_1)} \left[\sum_{l=0}^{k_1} \sum_{m=0}^{k_1} H_{2D}(l, m) \cos(w_1 l) \cdot \cos(w_2 m) \right] \quad (105)$$

by using the generalized McClellan Transformation (chapter II).

$$m(w_2, w_1) = \sum_{i=0}^a \sum_{g=0}^b c(i, g) \cos(iw_1) \cos(gw_2) \quad (3)$$

The filter designer can elect to use either the first order transformation ($a = b = 1$) or the second order transformation ($a = b = 2$).

Although the program can design lowpass, highpass, bandpass, bandstop, all-pass, and multiband digital filters; it should be kept in mind that only the shape of a single contour in the two-dimensional frequency plane can be closely approximated by the design program (chapter II). Therefore, the program lends itself to the design of lowpass, highpass, and all-pass filters.

The design program that was developed is an interactive program. This was necessary for two reasons. First, there is no guarantee that the McClellan Transformation will produce a well-defined mapping from the one-dimensional frequency axis, w , to the two-dimensional frequency plane, w_2, w_1 . If the mapping is ill-defined, filter designer intervention is necessary. Second, the one-dimensional filter

order necessary to design a one-dimensional filter with desired characteristics (band error, etc.) is not known in advance. This requires experimentation on the part of the filter designer.

The required inputs to the design program are a set of points that defines the contour in the two-dimensional frequency plane that is to be approximated, and the parameters necessary to design the one-dimensional digital filter. This filter will be transformed into a two-dimensional filter. The design program will output the contour approximating function, the impulse response samples of the one-dimensional prototype filter, a graph of the magnitude of the two-dimensional frequency response in the first quadrant (the frequency response has quadrantal symmetry), and the unique two-dimensional impulse response samples. As options, the program will output sets of points for any two-dimensional contours, a Calcomp plot of the one-dimensional filter's magnitude versus frequency characteristic, a Calcomp plot of the two-dimensional contours, and a Calcomp plot of the two-dimensional filter's magnitude versus frequency characteristic.

Figure 2 shows design times for typical two-band filters designed on the CDC 6600 computer (Fortran IV compiler). The filters were designed with one pass through the program. The scaling routine and the first order McClellan Transformation were used. The design times will be roughly double if the second order McClellan Transformation

is used. NFILT represents the order of the one-dimensional filter that is transformed into the desired two-dimensional filter during the design process.

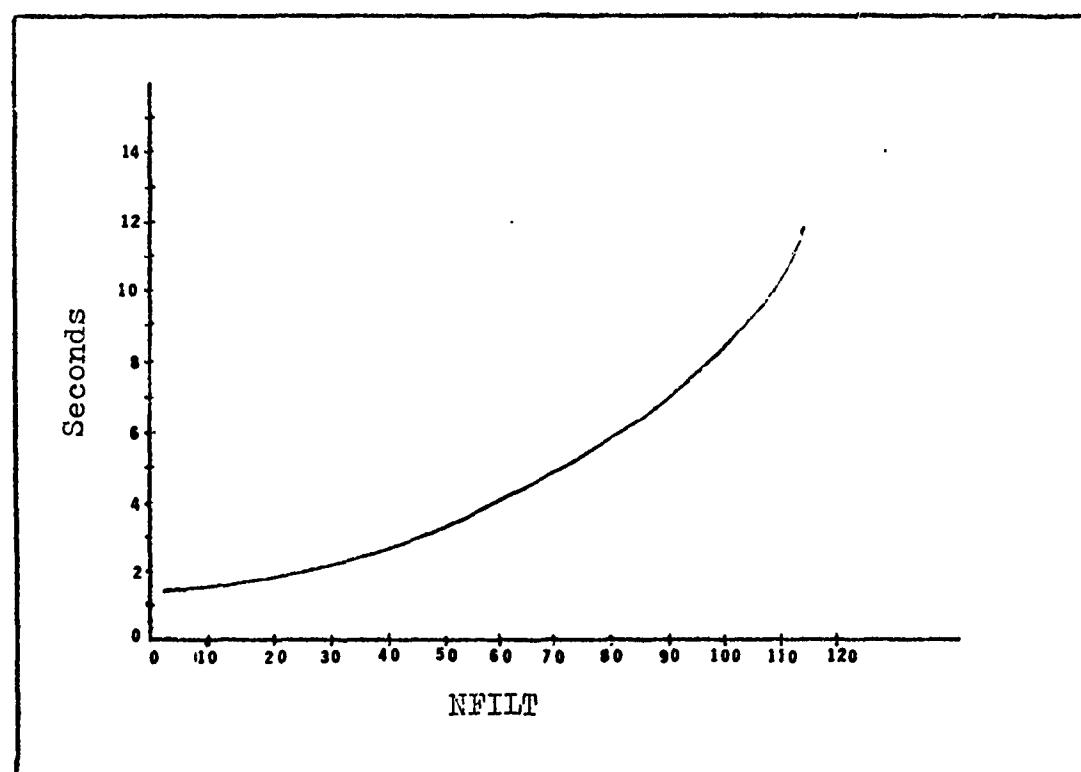


Fig 2. Typical Design Times for Two-Band Filters

Figure 3 shows a high level flow chart of the design program. It shows the seven separate programs that make up the filter design program and their subprograms. Each program is resident in its own overlay. The seven separate programs are executed in the order shown with the top program being executed first. Also shown are the variables that are transmitted between the programs (variable names in parentheses). An outbound arrow indicates that the variable

originates in the overlay and an in round arrow indicates that the variable is input to the overlay. All variables are transmitted between overlays by labeled common storage areas.

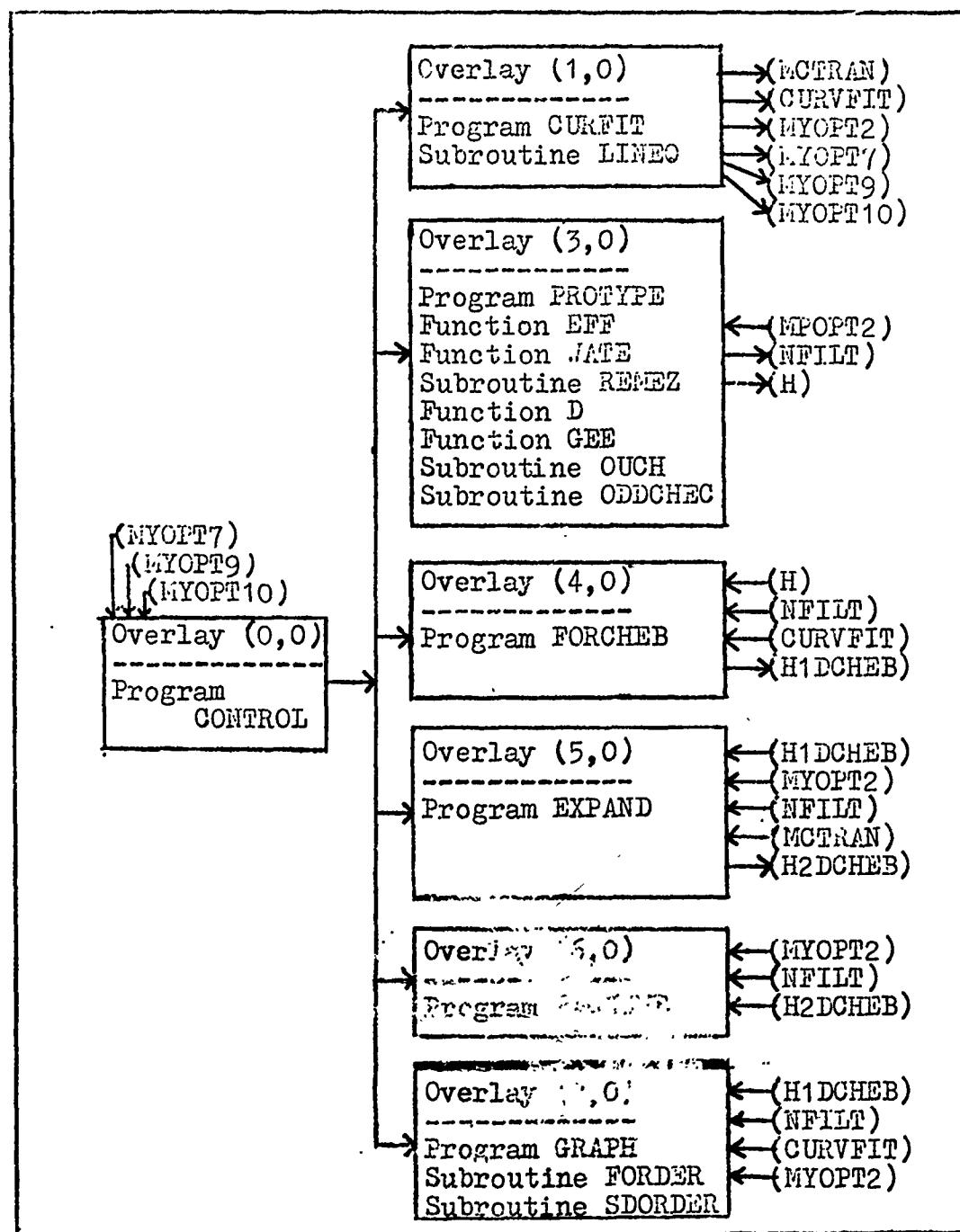


Fig 3. High Level Flow Chart of the Design Program

Program CONTROL

The main function of program CONTROL is to call the other six programs that make up the two-dimensional filter design program. It also allows the user to interactively terminate execution of the design program at several places if he wishes. A high level flow chart of program CONTROL is shown in figure 4.

Program CURFIT

The primary function of program CURFIT is to calculate the constants of the McClellan Transformation. It also tests to see whether or not a well-defined mapping will be produced by the transformation.

The first thing that program CURFIT does is calculate the $c(i,g)$'s of equation (3). This is done by solving a set of simultaneous linear equations of the form $A \cdot CURVFit = B$. The elements of the arrays A and B depend on w , a set of points defining a contour in the w_2, w_1 plane, and a set of constraints. Subroutine LINWQ solves the set of equations and returns the $c(i,g)$'s in the CURVFit array.

After the $c(i,g)$'s are calculated, equation (22) is evaluated on a grid of points in the w_2, w_1 plane in order to see if the mapping from the w -axis to the w_2, w_1 plane will be well-defined. If the mapping is ill-defined, the user can elect (interactively) to start over, choose a new set of constraints and repeat the approximation process, shift and scale, or terminate the design program. If shifting and

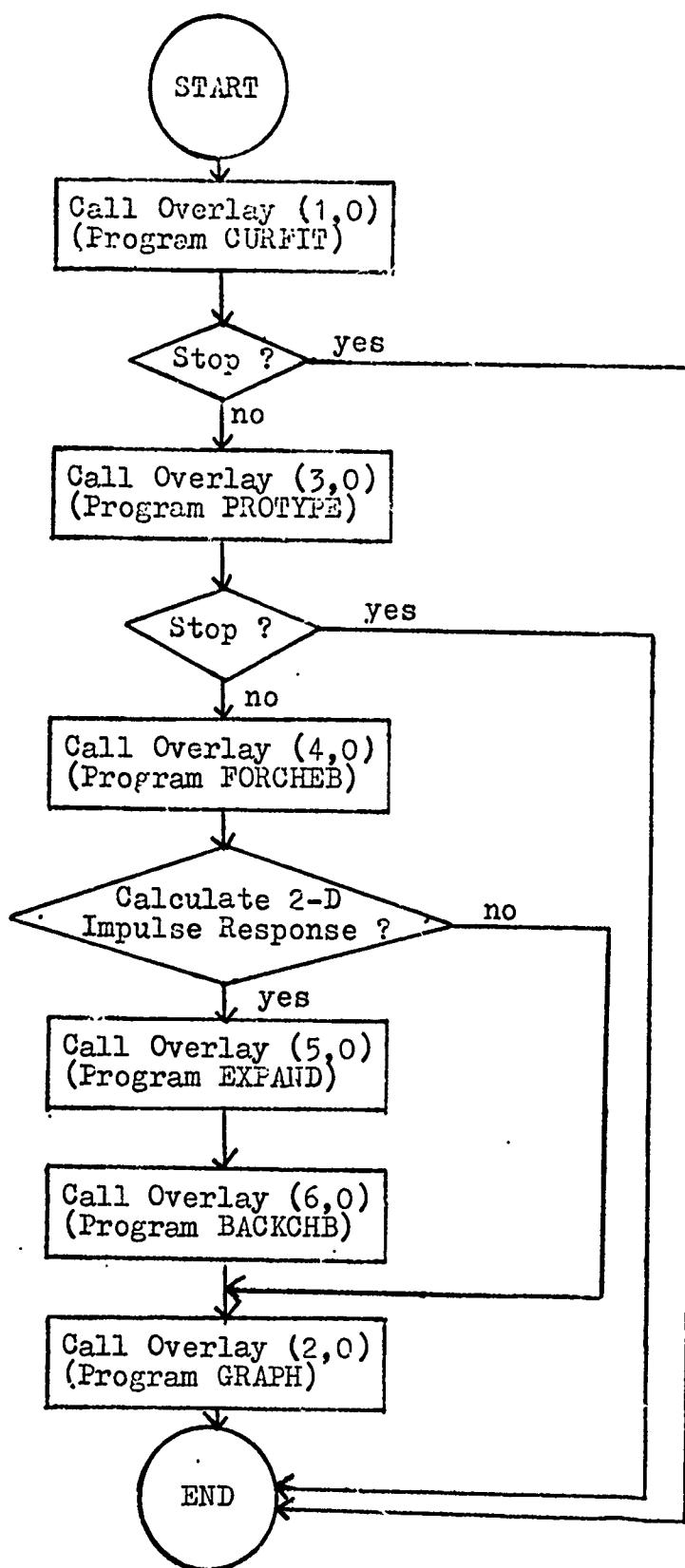


Fig 4. Flowchart of Program CONTROL

scaling is chosen, the program calculates the new $c(i,g)$'s by equations (26) through (40). F_{\max} and F_{\min} (equations (26) and (27)) are calculated by evaluating equation (3) on a grid of points in the w_2, w_1 plane.

Once a set of $c(i,g)$'s is calculated that produces a well-defined mapping, equation (3) is converted from the form $\sum_{i=0}^a \sum_{g=0}^b c(i,g) \cos(iw_1) \cos(gw_2)$ to the form

$$\sum_{i=0}^a \sum_{g=0}^b MCTRAN(i,g) [\cos(w_1)]^i [\cos(w_2)]^g$$
 by using equations

(41) through (44). A high level flowchart of program CURFIT is shown in figure 5.

Program PROTOTYPE

The primary function of program PROTOTYPE is the calculation of the impulse response of the one-dimensional filter that will be transformed into the two-dimensional filter. Program PROTOTYPE is a modified version of the computer program described in reference 8. There are several good one-dimensional filter design programs available in the literature. The program described in reference 8 was chosen because it is well documented. Program PROTOTYPE uses the Remez exchange algorithm to design one-dimensional filters with minimum weighted Chebyshev error in the filter bands. For a complete description, including detailed flowcharts and design examples, of the one-dimensional filter design program used in PROTOTYPE see reference 8. Program PROTOTYPE outputs the one-dimensional impulse response and a summary

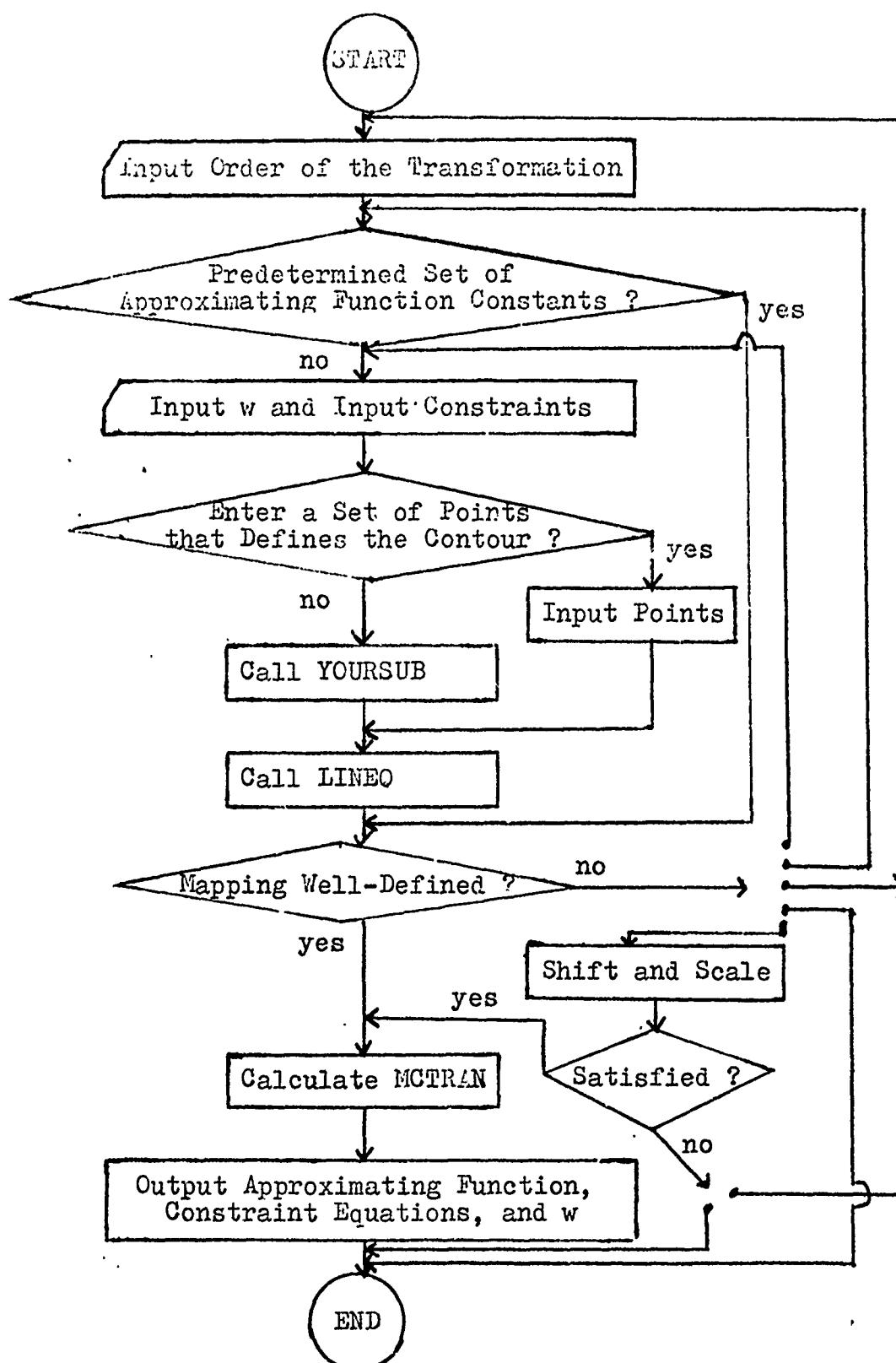


Fig 5. High Level Flowchart of Program CURFIT

of design parameters input by the user.

Program FORCHEB

The primary function of program FORCHEB is to perform the forward Chebyshev recursion in one variable. The first thing that program FORCHEB does is convert the one-dimensional impulse response, $h(n)$, to $H1D(n)$ (equation (112)) by using equations (113) through (115). Then the program generates the D array of equation (62) (called DAV in the program) and converts $H1D$ (equation(112)) to $H1DCHEB$ (equation (117)) by performing the forward Chebyshev recursion in one variable (equation (62)). Finally FORCHEB calculates the magnitude of the two-dimensional frequency response by using equation (118) at 121 points in the w_2, w_1 plane and outputs these values as a graph. The graph allows the user to see roughly where each filter band lies in the two-dimensional frequency plane. A high level flowchart of program FORCHEB is shown in figure 6.

Program EXPAND

Program EXPAND converts equation (119) to equation (121) by performing the required multiplications and summations. Essentially all of the terms of equation (120) are generated and then added together. The multiplication routine multiplies like a person would with paper and pencil. For each P (equation (120)), every term of the multiplier multiplies every term of the multiplicand and the resulting partial sums are added together to generate each P.

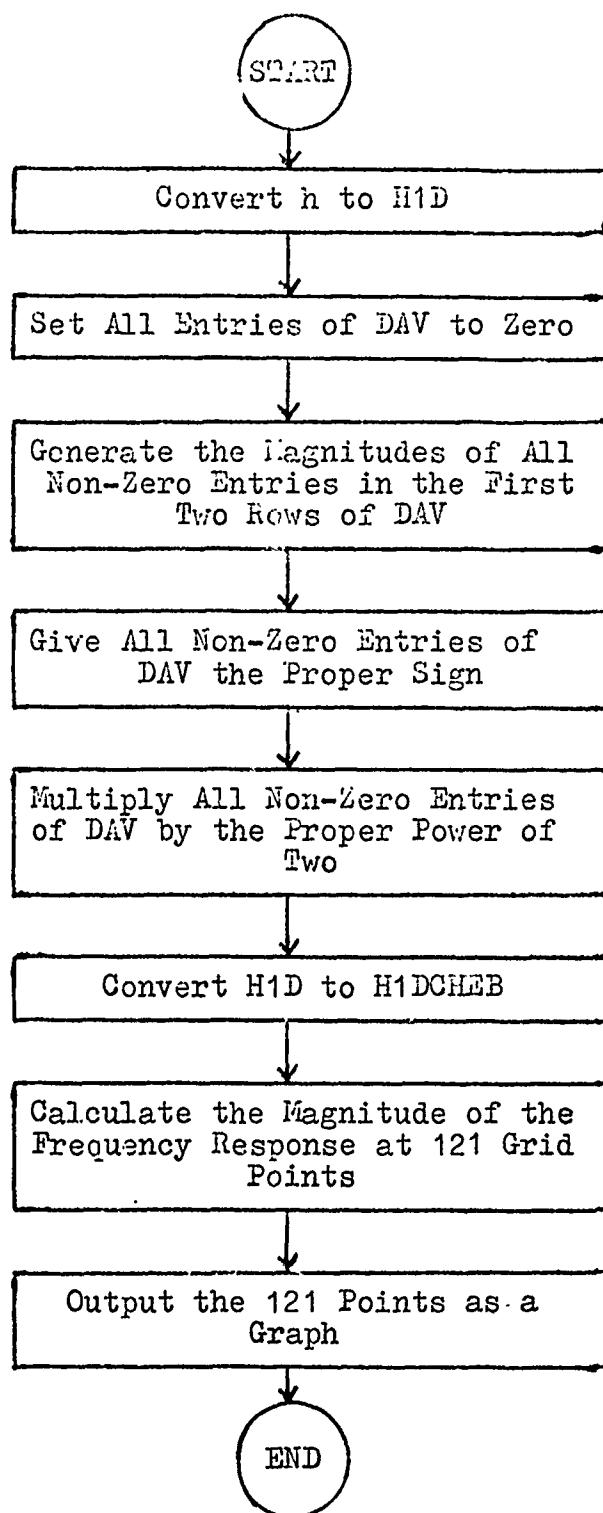


Fig 6. High Level Flowchart of Program FORCHEB

Program BACKCHB

The primary function of program BACKCHB is to perform the backward Chebyshev recursion in two variables. The first thing BACKCHB does is to generate the array C of equation (85) (called CEC in the program). Then by using equation (85) twice, H2DCHEB (equation (121)) is converted to H2D (equation (121)). Finally H2D is converted into the two-dimensional impulse response by using equations (106) through (109) solved for H2D. The output of BACKCHB consists of a list of the unique two-dimensional impulse response samples plus the symmetry formulas necessary to determine the non-unique impulse response samples. The impulse response constitutes the filter design as defined in this investigation. The symmetry formulas are based on equations (103) and (104). A high level flowchart of program BACKCHB is shown in figure 7.

Program GRAPH

Program GRAPH generates most of the plots produced by the design program. All plots are optional and include a Calcomp plot of the one-dimensional filter's magnitude versus frequency characteristic and a Calcomp plot of the two-dimensional contours. These plots give the user a visual means of judging how close the designed two-dimensional filter is to the desired two-dimensional filter. The program will also generate sets of points for any two-dimensional contours. This allows the user to determine the

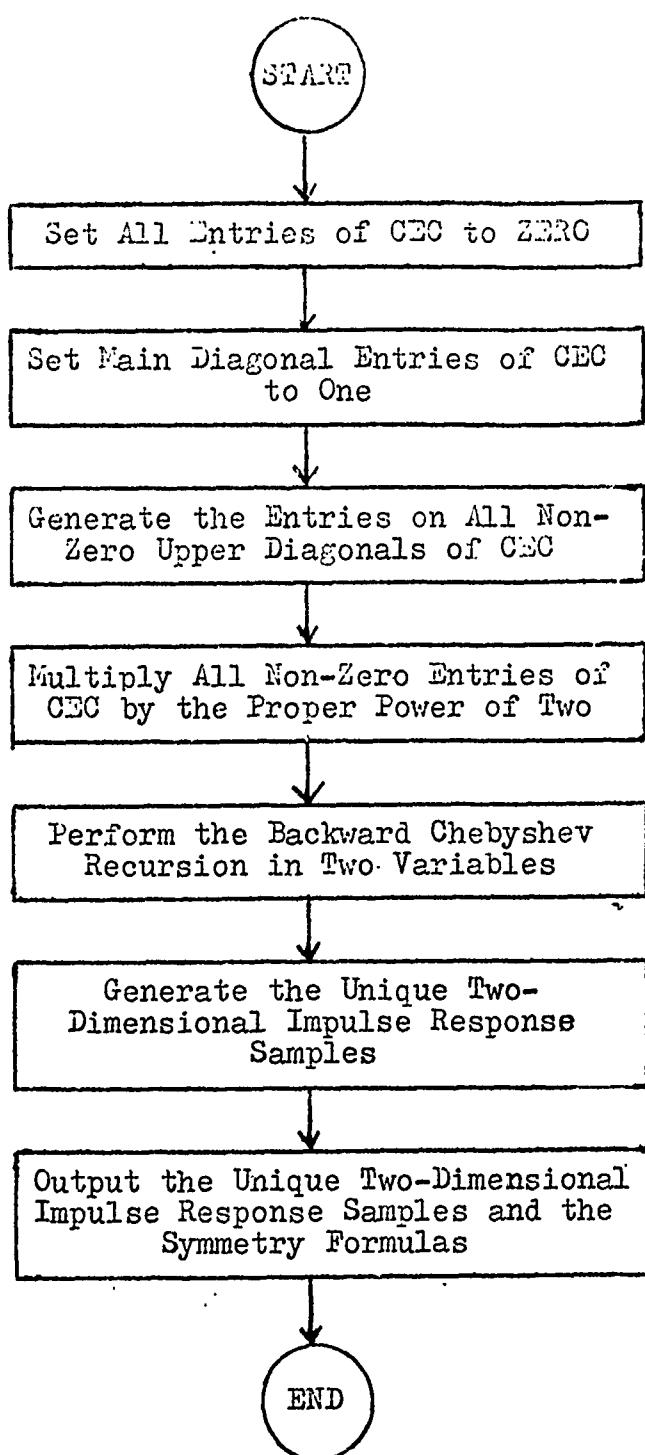


Fig 7. High Level Flowchart of Program BACKCHB

exact location of the band edges of the two-dimensional filter. Finally, the program creates the file (tape2) necessary for the Calcomp plot of the two-dimensional filter's magnitude versus frequency characteristic. Another program is used to generate the actual points and plot the three-dimensional figure. The three-dimensional figure gives the user a pictorial representation of the magnitude characteristic of the two-dimensional filter.

Subroutine FORDER generates the points for the plot of the two-dimensional contours when the first order McClellan Transformation is used for the two-dimensional filter design. This is done by using the equation of the first order McClellan Transformation solved for w_1 . A curve is plotted for the eleven values of w from 0.0pi to 1.0pi at 0.1pi intervals. This spacing gives the plot an uncluttered appearance.

Subroutine SDORDER generates the points for the plot of the two-dimensional contours when the second order McClellan Transformation is used for the two-dimensional filter design. This is done by using the equation of the second order McClellan Transformation solved for w_1 . This equation is a quadratic and can have two solutions for each w_1 . If two solutions exist then a curve is plotted for the eleven values of w from 0.0pi to 1.0pi at 0.1pi intervals for each solution (22 curves total). Again, this spacing gives the plot an uncluttered appearance. The eleven curves for one solution have a "+" at one or both ends of each curve and the eleven

curves for the other solution have an "x" at one or both ends of each curve. If the quadratic has only one solution for each w_1 , then only one set of eleven curves is plotted. A high level flowchart of program GRAPH is shown in figure 8.

Summary

This chapter has presented a brief description of the two-dimensional filter design program developed in this investigation. Each of the seven separate programs that make up the filter design program were described.

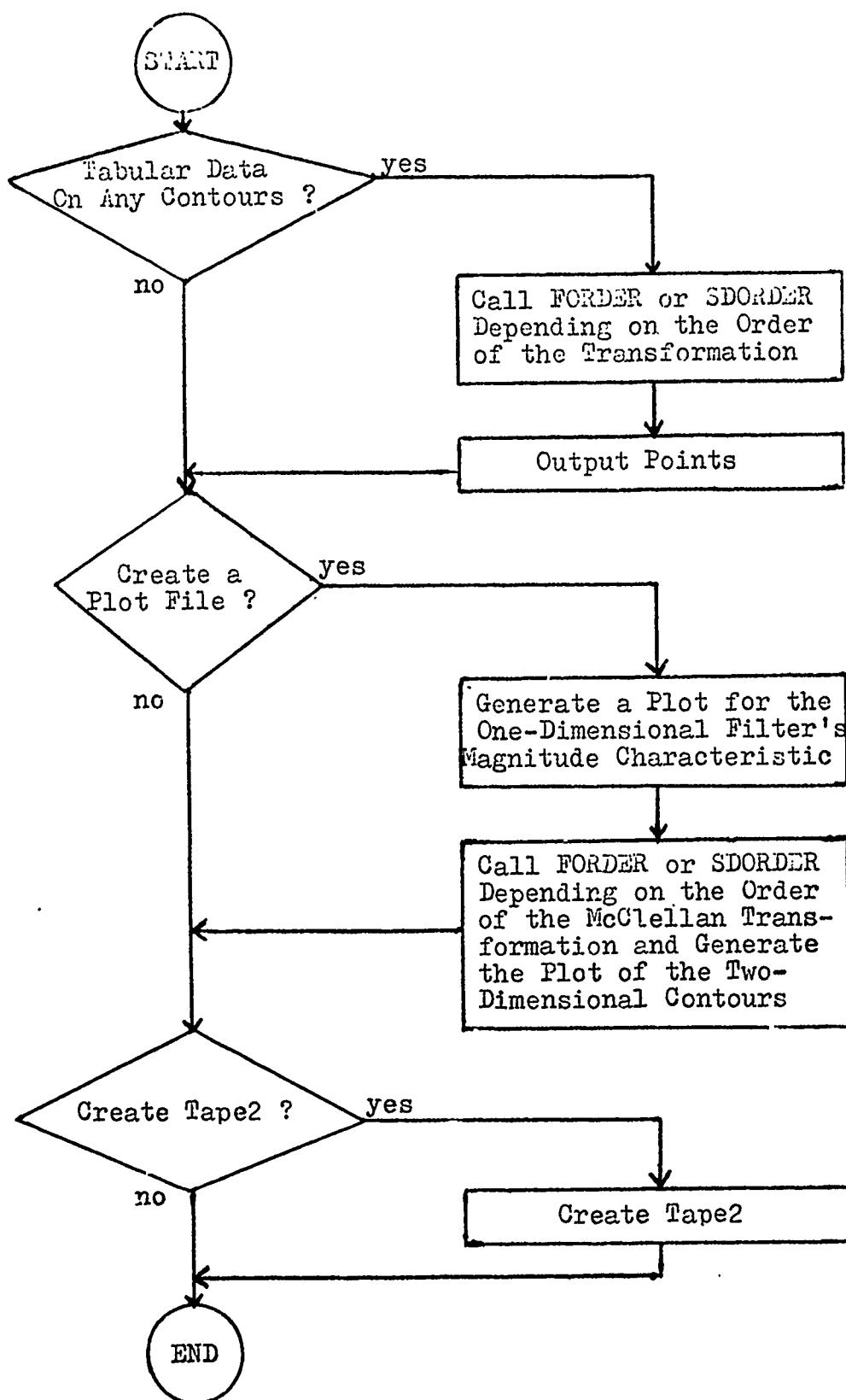


Fig 8. High Level Flowchart of Program GRAPH

VI Design Results

This chapter presents the results from five typical two-dimensional filter designs using the computer program developed in this investigation. A wide variety of contour shapes were used in the filter designs. The designs clearly illustrate the flexibility of the transformation of variable technique in designing two-dimensional digital filters.

Filter Design 1

The first filter design utilized hyperbolic contours. Figure 9 shows a plot of the two-dimensional contours. The following is a summary of the data pertaining to the contours.

1. Transformation: second order
2. Equation defining the desired contour: $w_1 = .3^2/2w_2$
3. Approximating function constants:

A=.71797 B=.37441 C=.37443 D=-.49945 E=.12559
F=.12557 G=-.0930 H=-.0930 I=-.03255

Figure 10 shows a plot of the one-dimensional filter's magnitude versus frequency characteristic. The following is a summary of the design parameters.

1. Number of transition bands: 2
2. Transition band edge frequencies: $.4\pi, .5\pi; .7\pi, .8\pi$
3. Magnitude for each band: 0, 1, 0
4. Ratio of the band errors: 1, 10, 1
5. Deviation in each band: .006, .0006, .006

A one-dimensional filter order of 57 was required to meet the

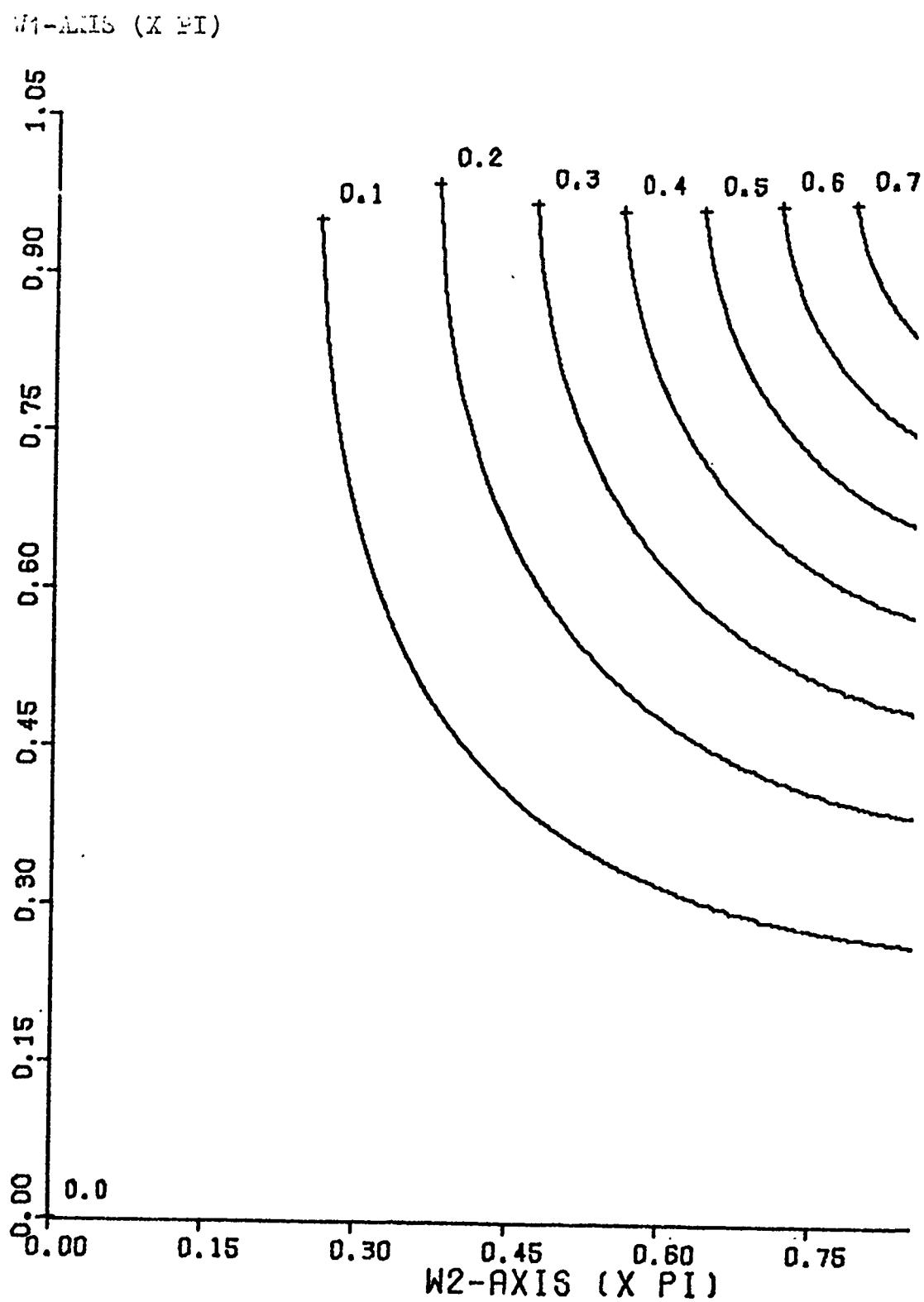


Fig 9. Filter Design 1: Two-Dimensional Contours

AMPLITUDE

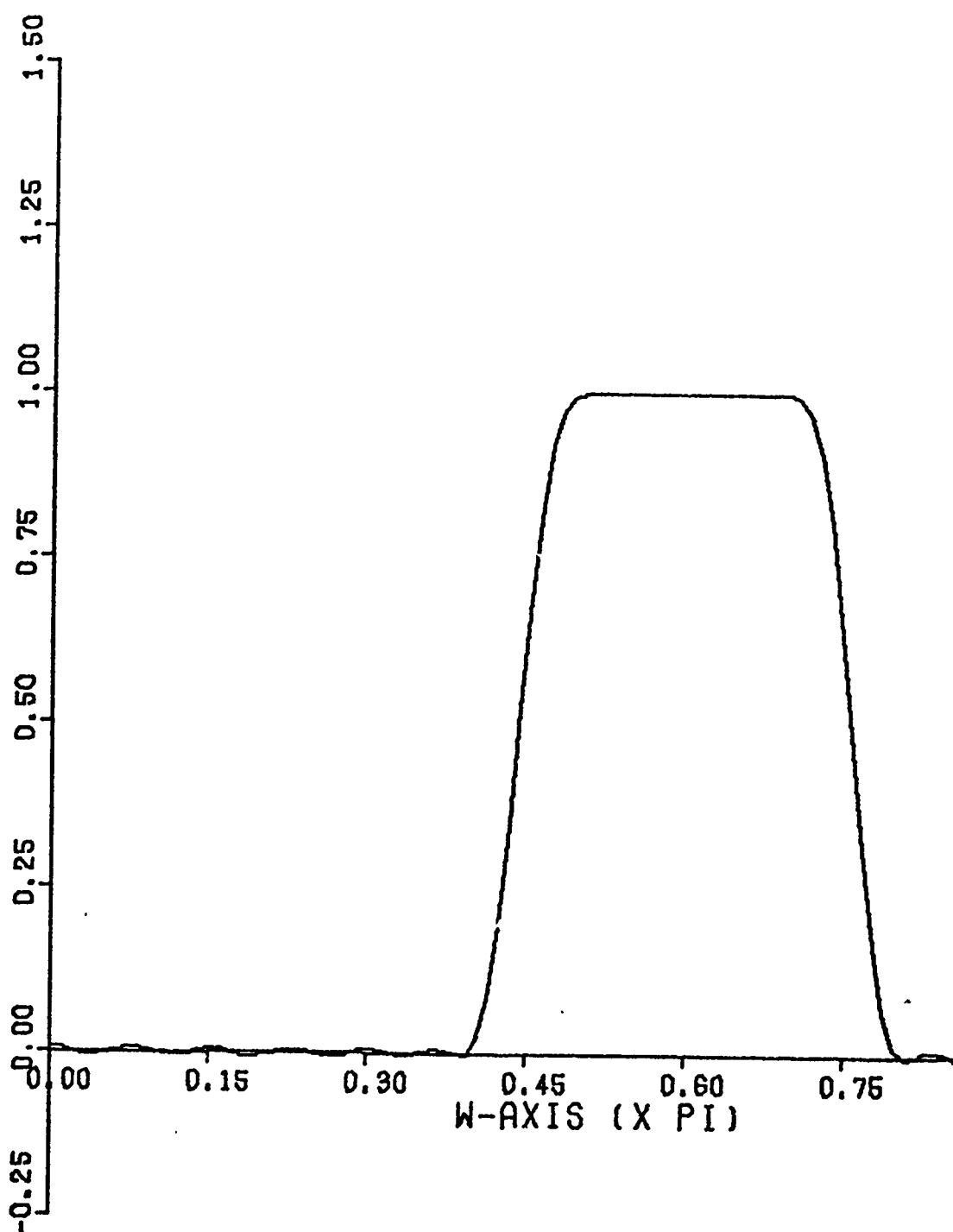


Fig 10. Filter Design 1: One-Dimensional Frequency Response

design parameter specifications. Figure 11 shows a plot of the two-dimensional filter's magnitude versus frequency characteristic.

Filter Design 2

The second filter design utilized triangular contours. Figure 12 shows a plot of the two-dimensional contours. The following is a summary of the data pertaining to the contours.

1. Transformation: second order
2. Set of points defining the desired contour (all coordinates times pi): (0,0), (.1,.1), (.2,.2), (.3,.3), (.4,.4), (.5,.3), (.6,.2), (.7,.1), (.8,0)
3. Approximating function constants:

$A=-.72211$ $B=-.37665$ $C=-.25873$ $D=.35644$ $E=-.2651$
 $F=-.09590$ $G=.10787$ $H=.19347$ $I=.06794$

Figure 13 shows a plot of the one-dimensional filter's magnitude versus frequency characteristic. The following is a summary of the design parameters.

1. Number of transition bands: 1
2. Transition band edge frequencies: $.87\pi$, $.9\pi$
3. Magnitude for each band: 0, 1
4. Ratio of the band errors: 1, 3
5. Deviation in each band: .12, .04

A one-dimensional filter order of 57 was required to meet the design parameter specifications. Figure 14 shows a plot of the resulting two-dimensional filter's magnitude versus frequency characteristic.

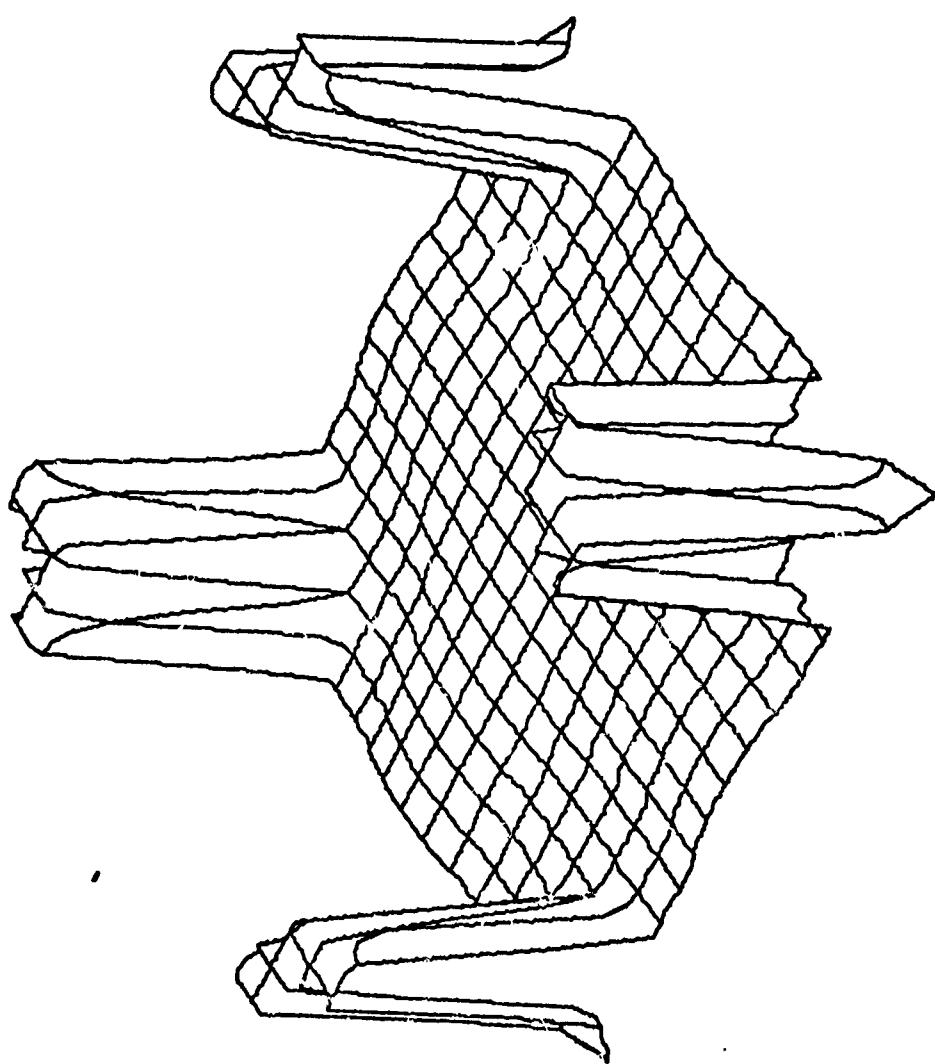


Fig 11. Filter Design 1: Two-Dimensional Frequency Response

W1-AXIS (X PI)

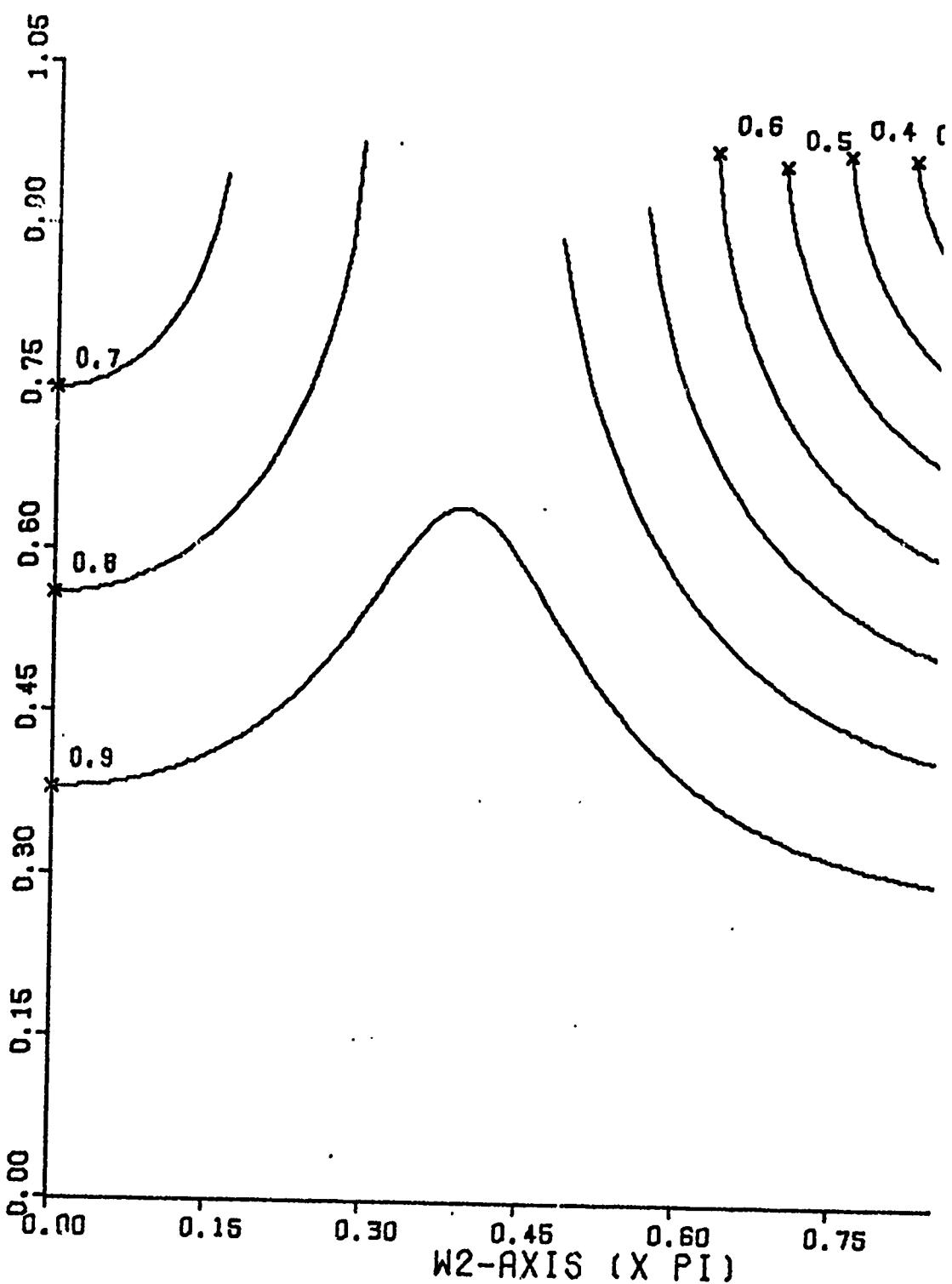


Fig 12. Filter Design 2: Two-Dimensional Contours

MAGNITUDE

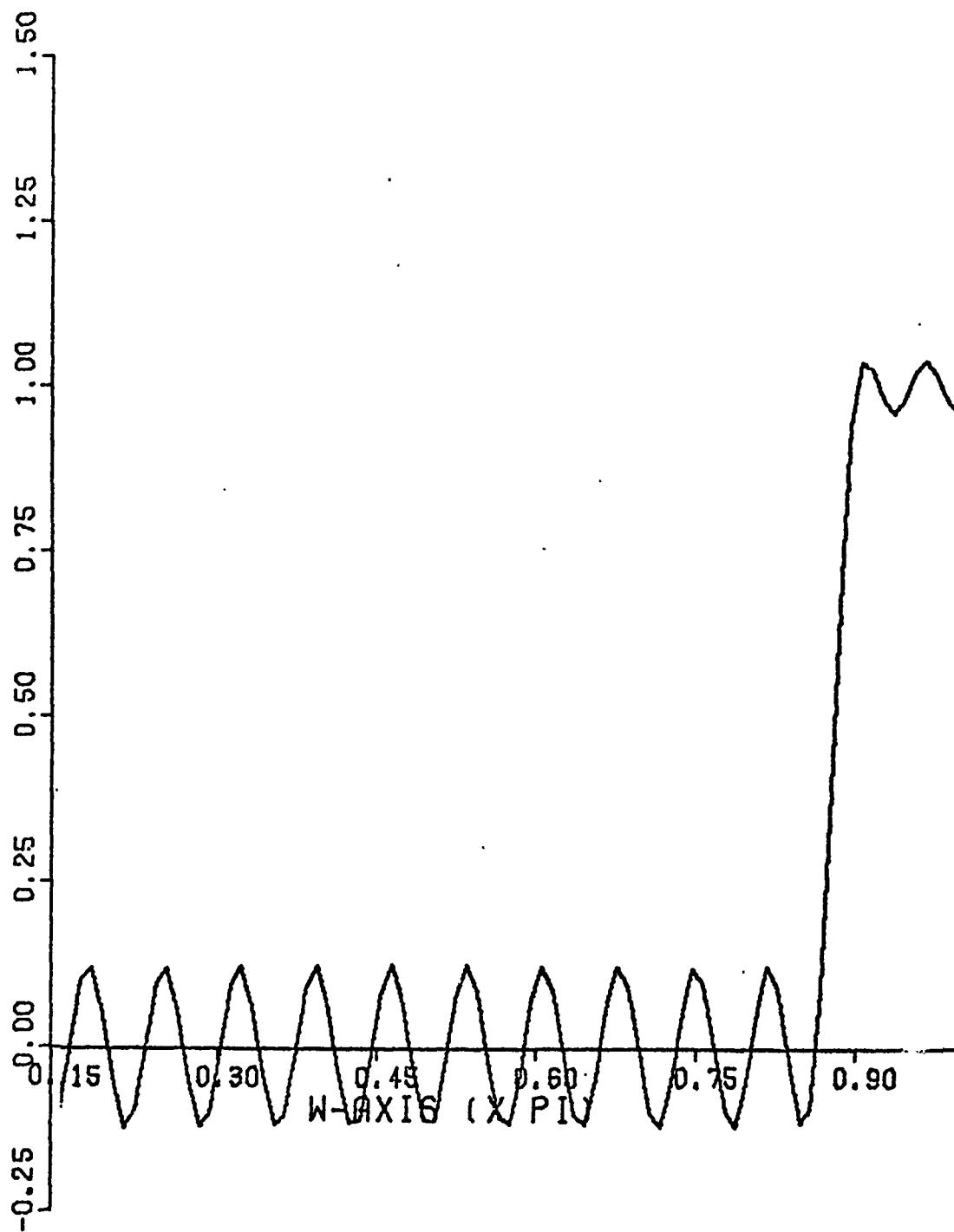


Fig 13. Filter Design 2: One-Dimensional Frequency Response

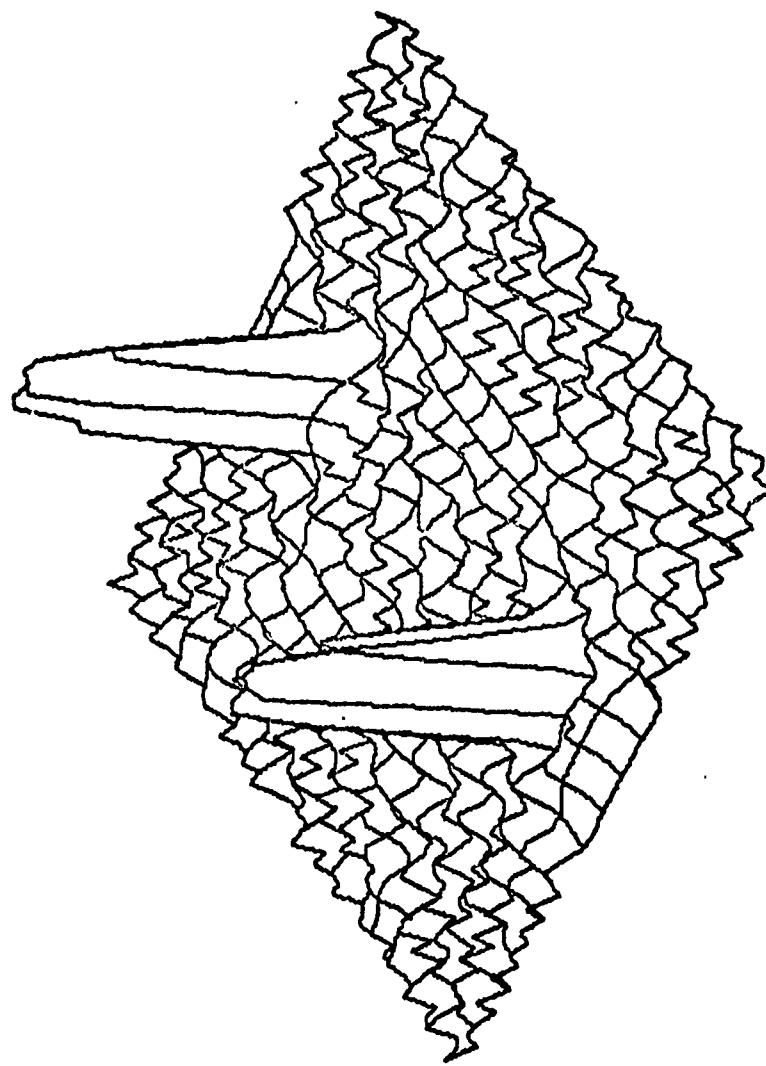


Fig 14. Filter Design 2: Two-Dimensional Frequency Response

Filter Design 3

The third filter design utilizes semi-circular contours. Figure 15 shows a plot of the two-dimensional contours. The following is a summary of the data pertaining to the contours.

1. Transformation: second order
2. Set of points defining the desired contour (all coordinates times pi): (.6,.6), (.55,.55), (.5,.5), (.45,.45), (.4,.4), (.45,.35), (.5,.3), (.55,.25)
3. Approximating function constants:

$$\begin{array}{llll} A=.03476 & B=-.30771 & C=.30901 & D=-.05277 \\ E=.05773 & F=-.05617 & G=.36402 & H=-.24187 \quad I=.05352 \end{array}$$

Figure 16 shows a plot of the one-dimensional filter's magnitude versus frequency characteristic. The following is a summary of the design parameters.

1. Number of transition bands: 1
2. Transition band edge frequencies: .6pi, .63pi
3. Magnitude for each band: 0, 1
4. Ratio of the band errors: 1, 1
5. Deviation in each band: .08, .08

A one-dimensional filter order of 57 was required to meet the design parameter specifications. Figure 17 shows a plot of the resulting two-dimensional filter's magnitude versus frequency characteristic.

Filter Design 4

The fourth filter design utilized right-angle contours. Figure 18 shows a plot of the two-dimensional contours. The following is a summary of the data pertaining to the contours.

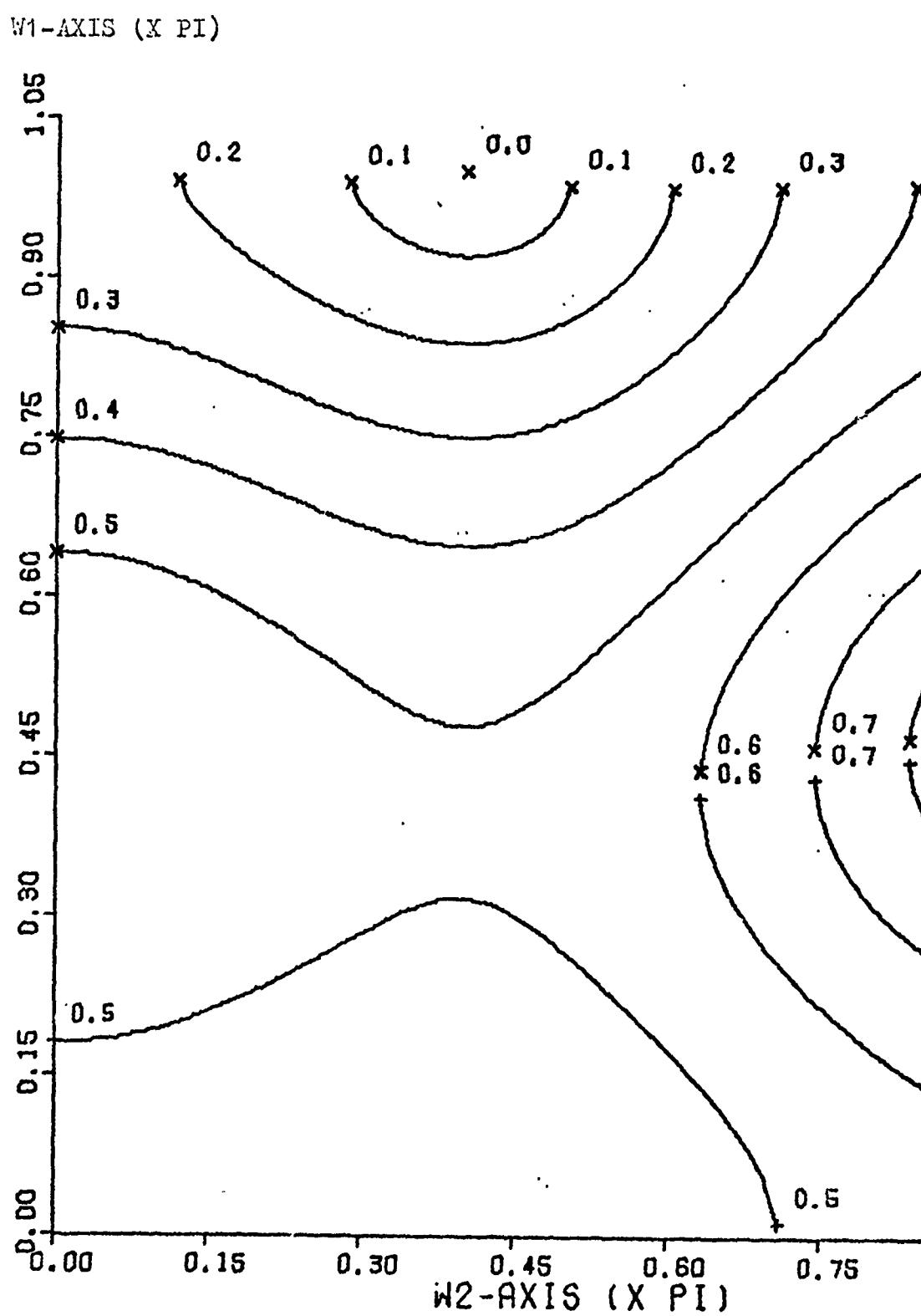


Fig. 15. Filter Design 3: Two-Dimensional Contours

MAGNITUDE

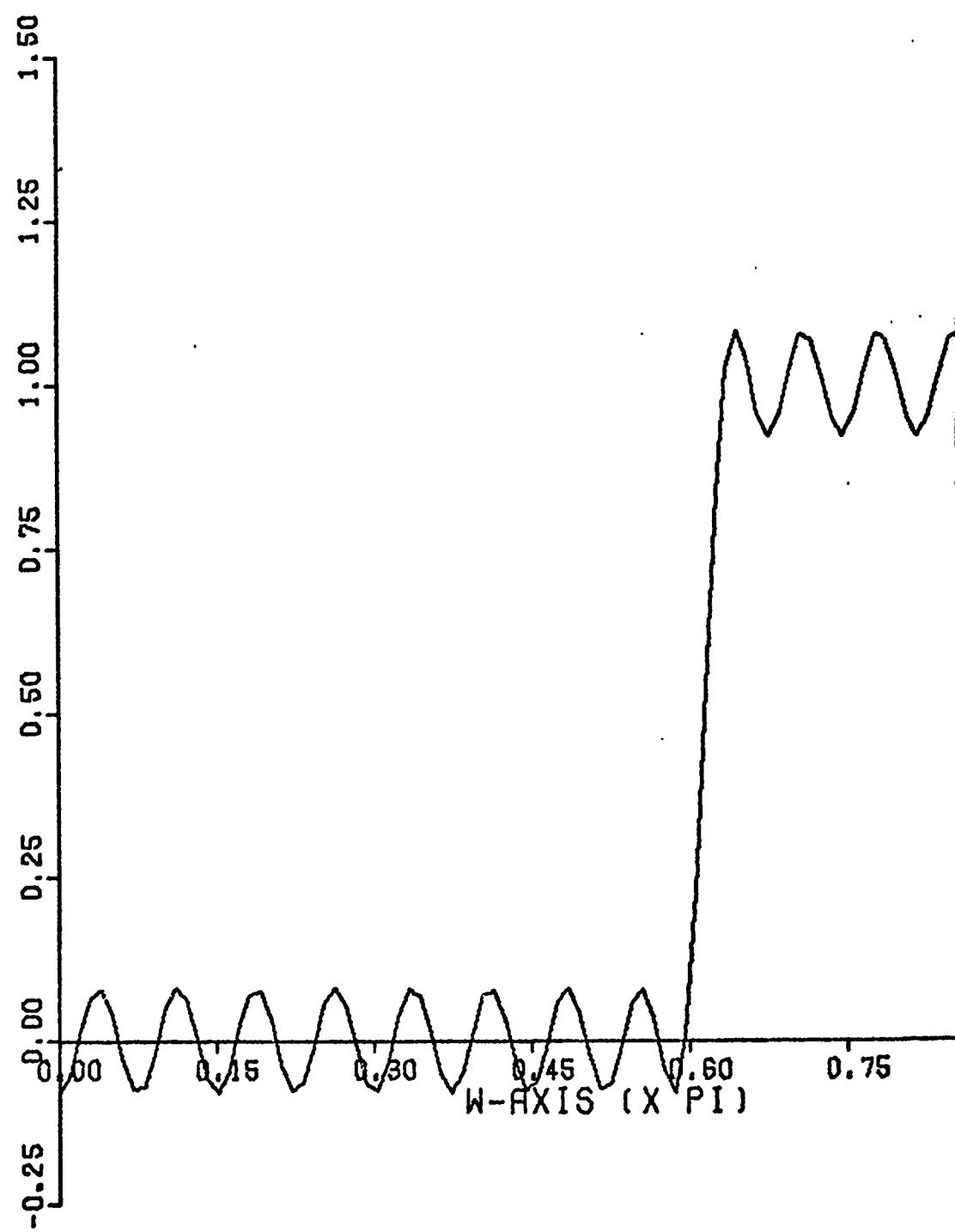


Fig 16. Filter Design 3: One-Dimensional Frequency Response

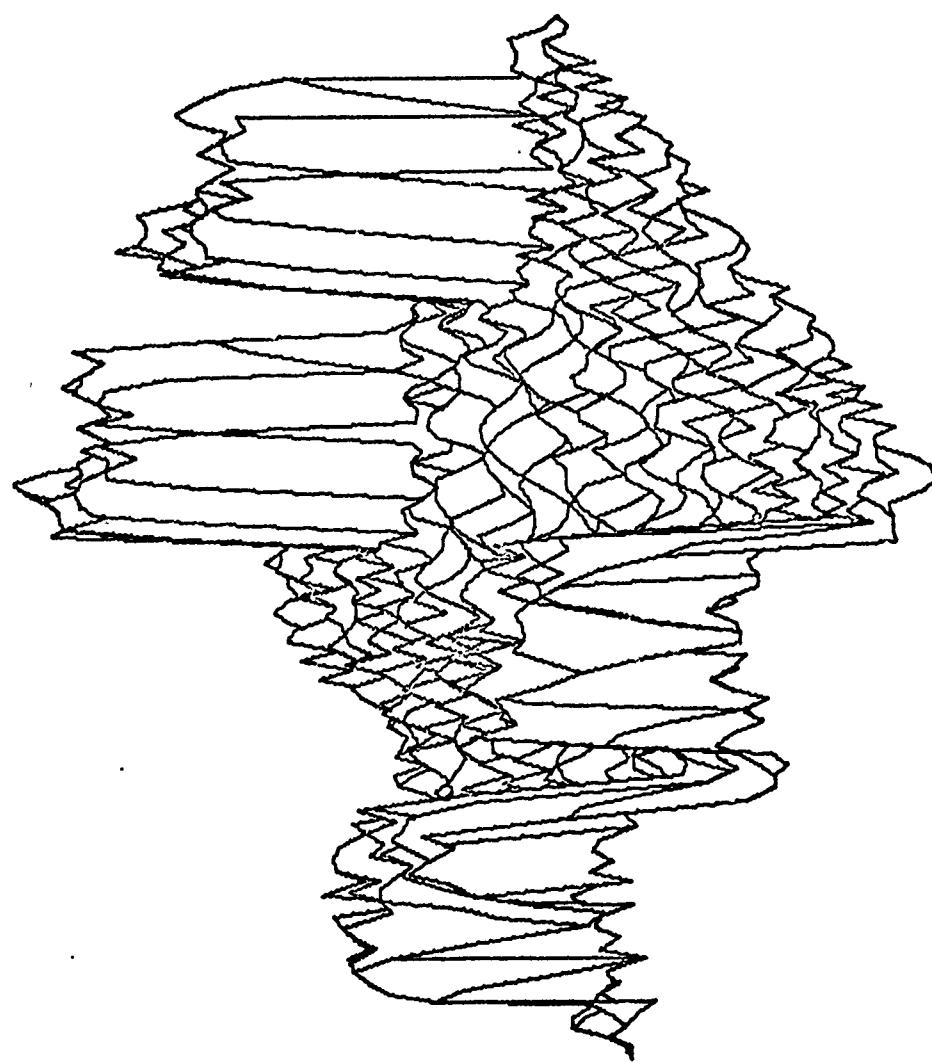


Fig 17. Filter Design 3: Two-Dimensional Frequency Response

W1-AXIS (X PI)

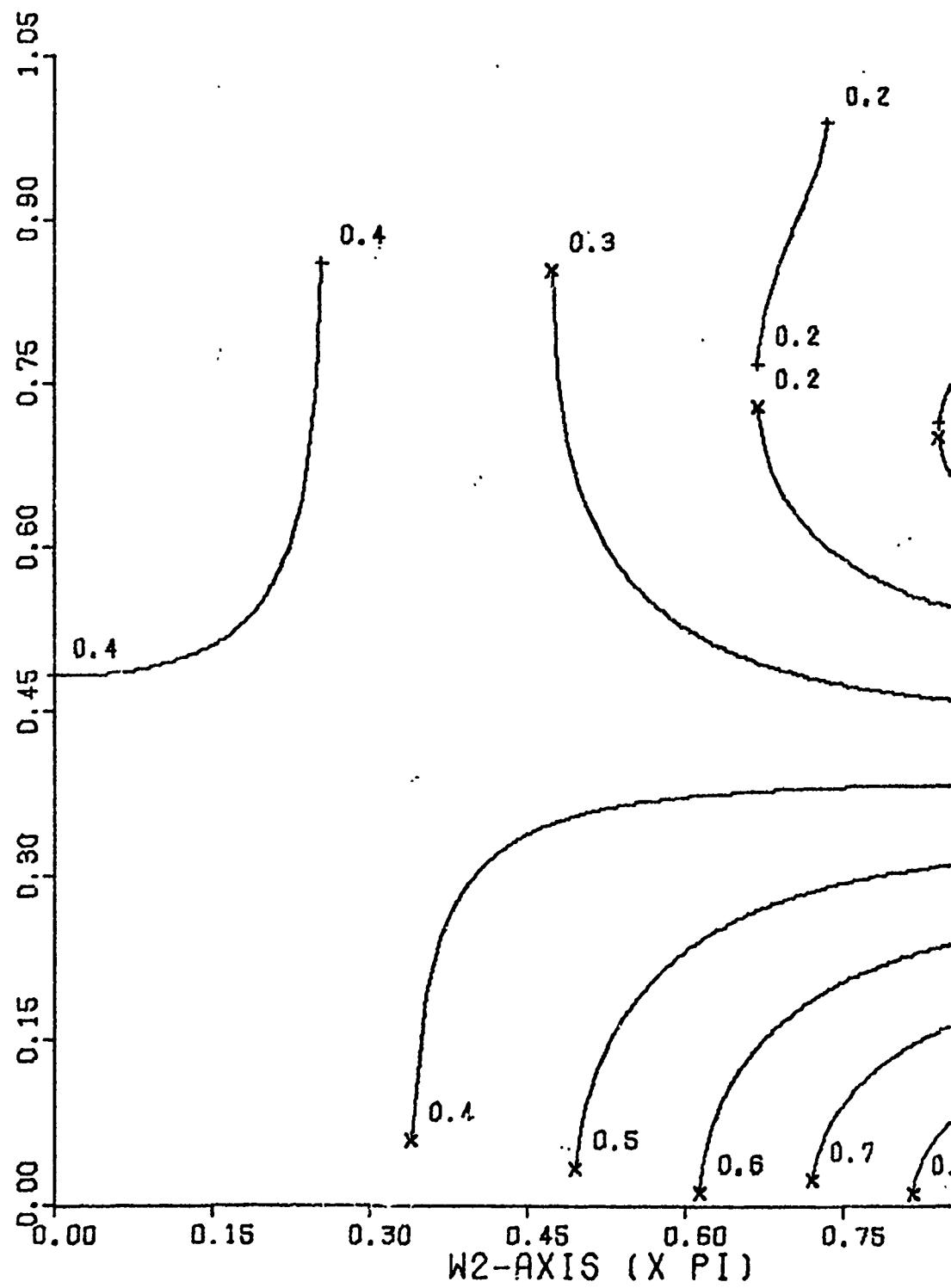


Fig 18. Filter Design 4: Two-Dimensional Contours

1. Transformation: second order
2. Set of points defining the desired contour (all coordinates times pi): (.3,.4), (.3,.6), (.3,.8), (.3,.9), (.3,1), (.4,.4), (.6,.4), (.8,.4), (.9,.4), (1,.4)

3. Approximating function constants:

A=.35752 B=-.34052 C=-.00607 D=.56701 E=-.02343
F=.20908 G=-.13863 H=-.03398 I=-.05095

Figure 19 shows a plot of the one-dimensional filter's magnitude versus frequency characteristic. The following is a summary of the design parameters.

1. Number of transition bands: 1
2. Transition band edge frequencies: .28pi, .3pi
3. Magnitude for each band: 1, 0
4. Ratio of the band errors: 2, 1
5. Deviation in each band: .1, .2

A one-dimensional filter order of 43 was required to meet the design parameter specifications. Figure 20 shows a plot of the resulting two-dimensional filter's magnitude versus frequency characteristic.

Filter Design 5

The fifth filter design utilizes circular contours. Figure 21 shows a plot of the two-dimensional contours. The following is a summary of the data pertaining to the contours.

1. Transformation: first order
2. Equation defining the desired contour:

$$w_1 = \sqrt{(.7\pi)^2 - w_2^2}$$

MAGNITUDE

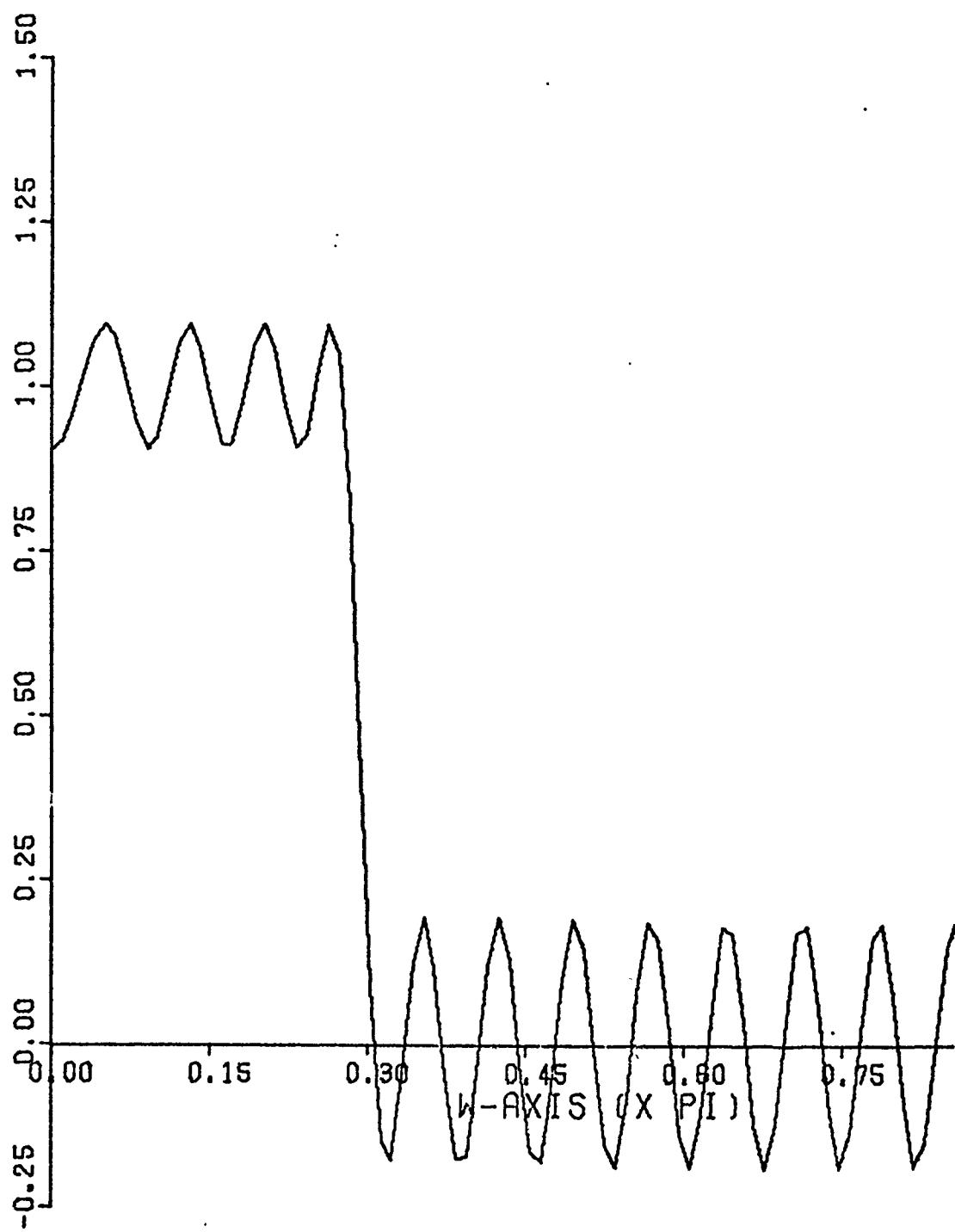


Fig 19. Filter Design 4: One-Dimensional Frequency Response

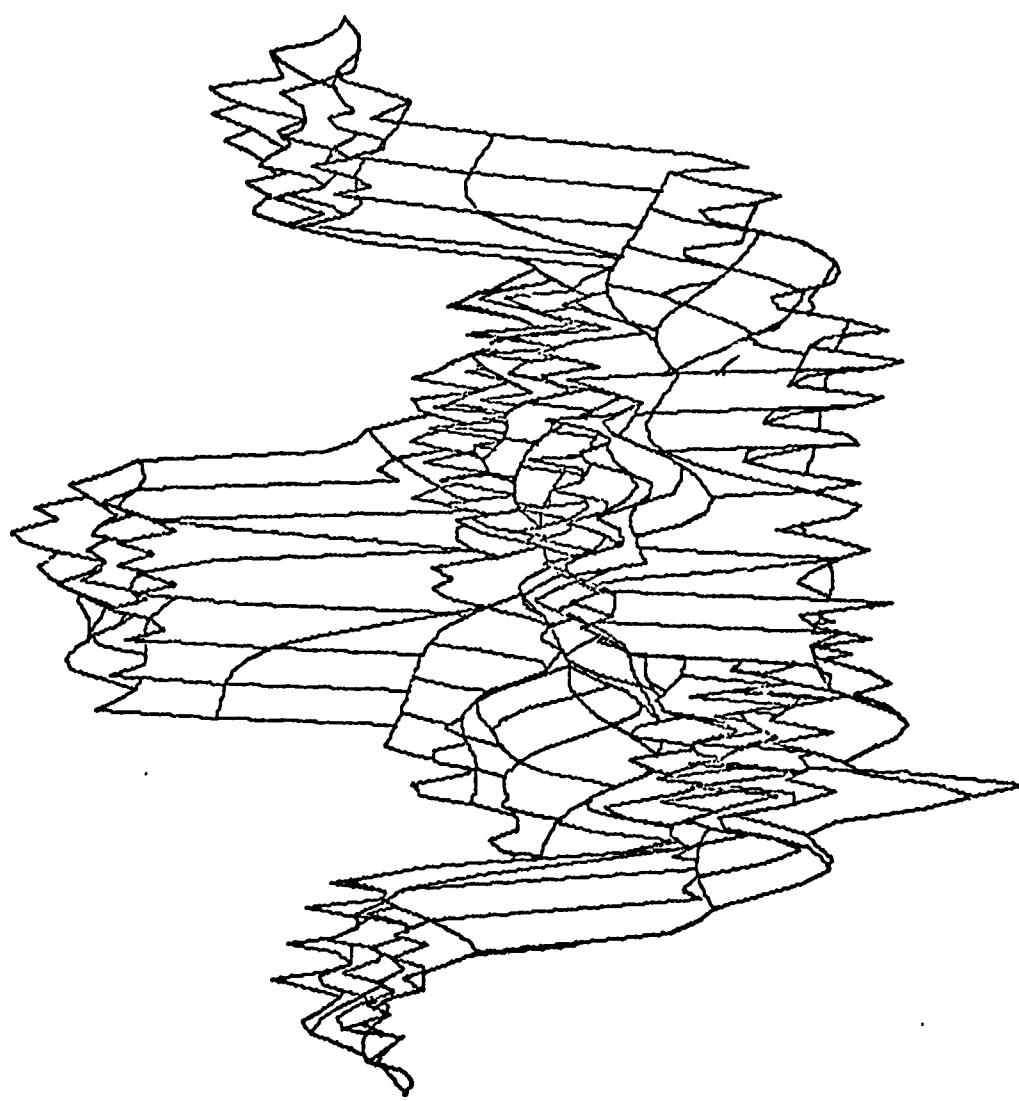


Fig 20. Filter Design 4: Two-Dimensional Frequency Response

W1-AXIS (X PI)

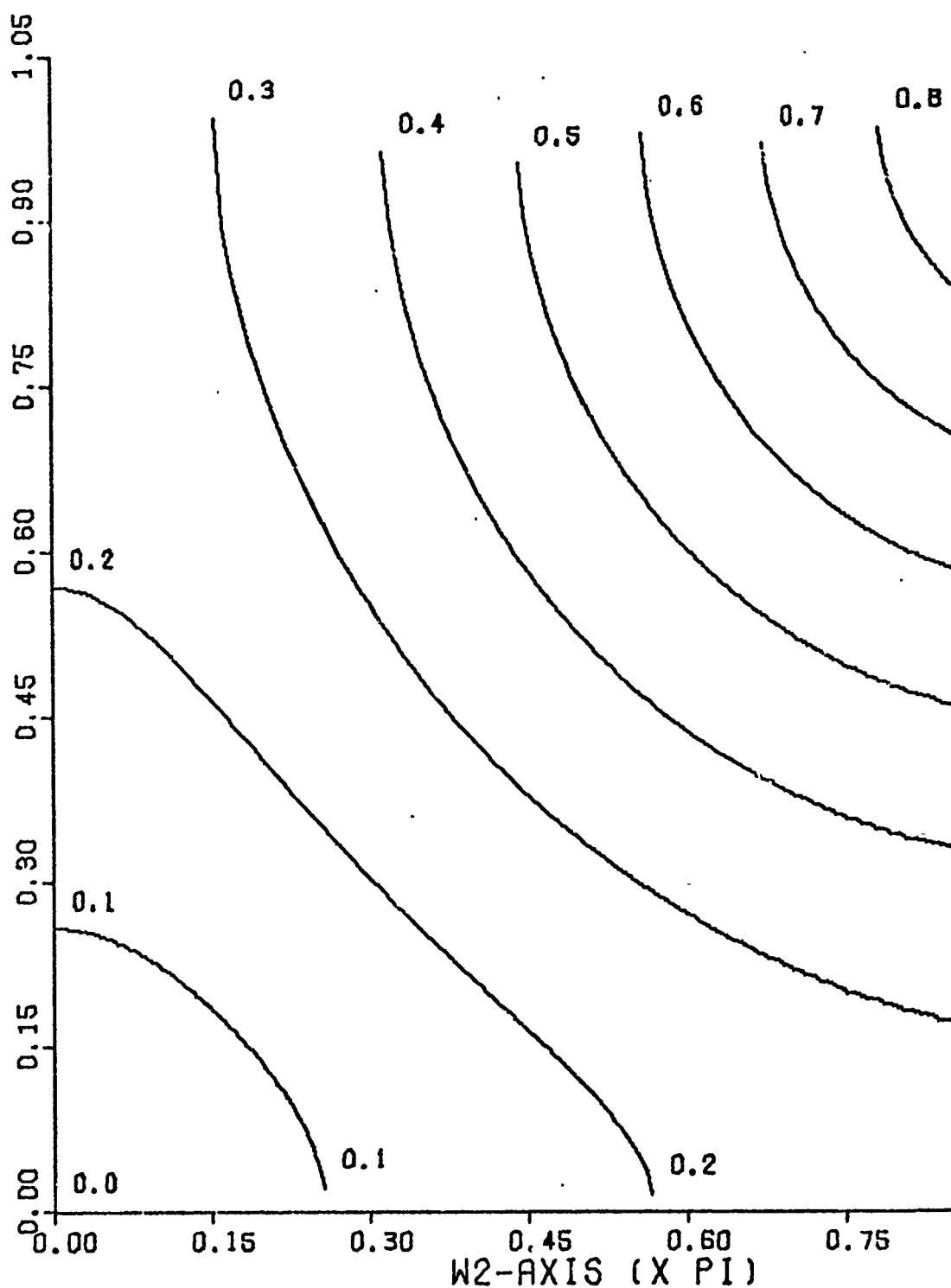


Fig 21. Filter Design 5: Two-Dimensional Contours

3. Approximating function constants:

$$A=.34220 \quad B=.5 \quad C=.5 \quad D=-.34220$$

Figure 22 shows a plot of the one-dimensional filter's magnitude versus frequency characteristic. The following is a summary of the design parameters.

1. Number of transition bands: 2
2. Transition band edge frequencies: $.15\pi, .2\pi; .4\pi, .5\pi$
3. Magnitude for each band: 0, 1, 0
4. Ratio of the band errors: 1, 2, 1
5. Deviation in each band: .14, .07, .14

A one-dimensional filter order of 31 was required to meet the design parameter specifications. Figure 23 shows a plot of the resulting two-dimensional filter's magnitude versus frequency characteristic.

Summary

In this chapter several filter designs obtained by using the two-dimensional digital filter design program developed in this investigation have been presented. The results demonstrate the flexibility of the transformation of variable technique for filter design. The filter designs presented show that this method can design two-dimensional filters with complex shapes.

MAGNITUDE

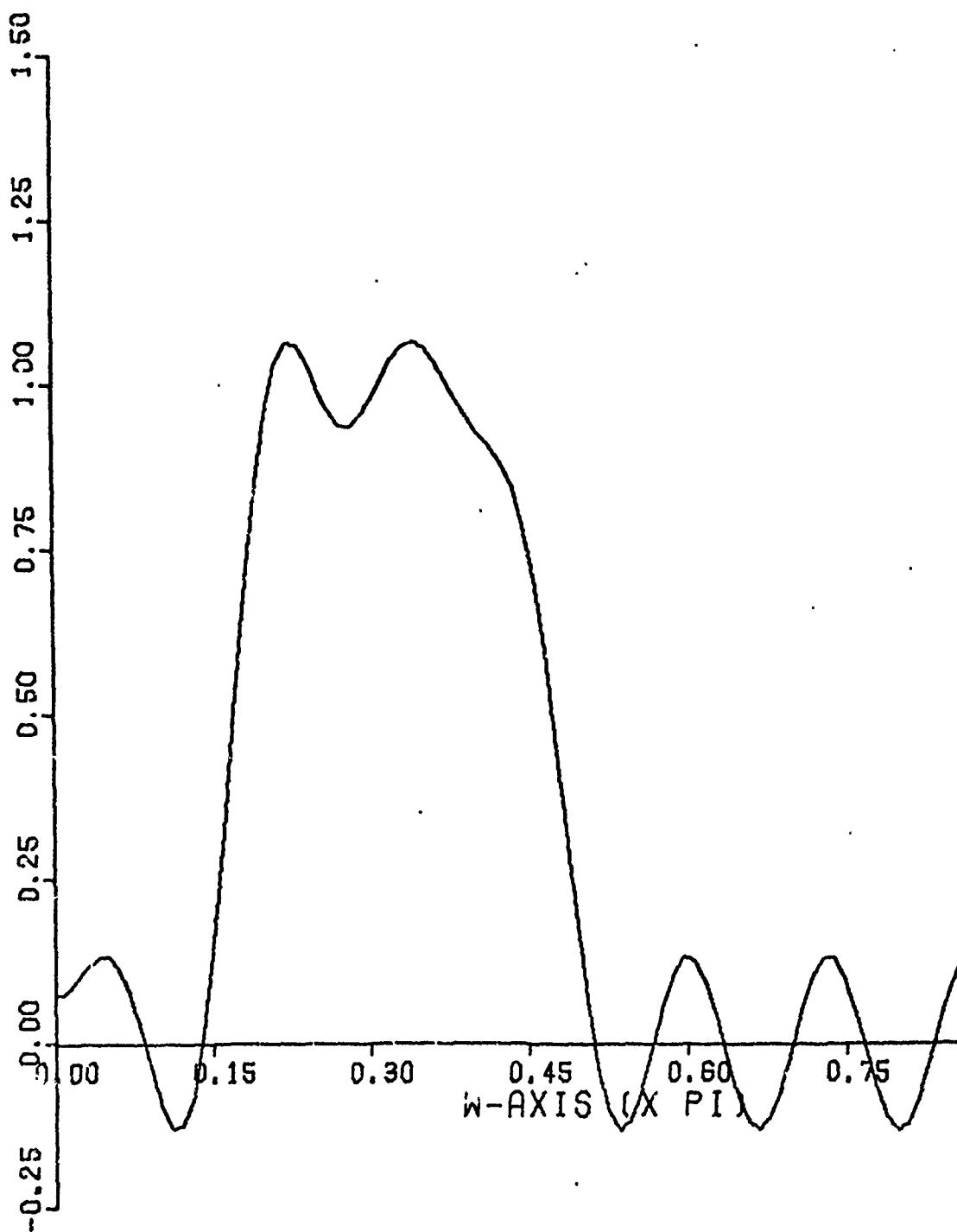


Fig 22. Filter 5: One-Dimensional Frequency Response

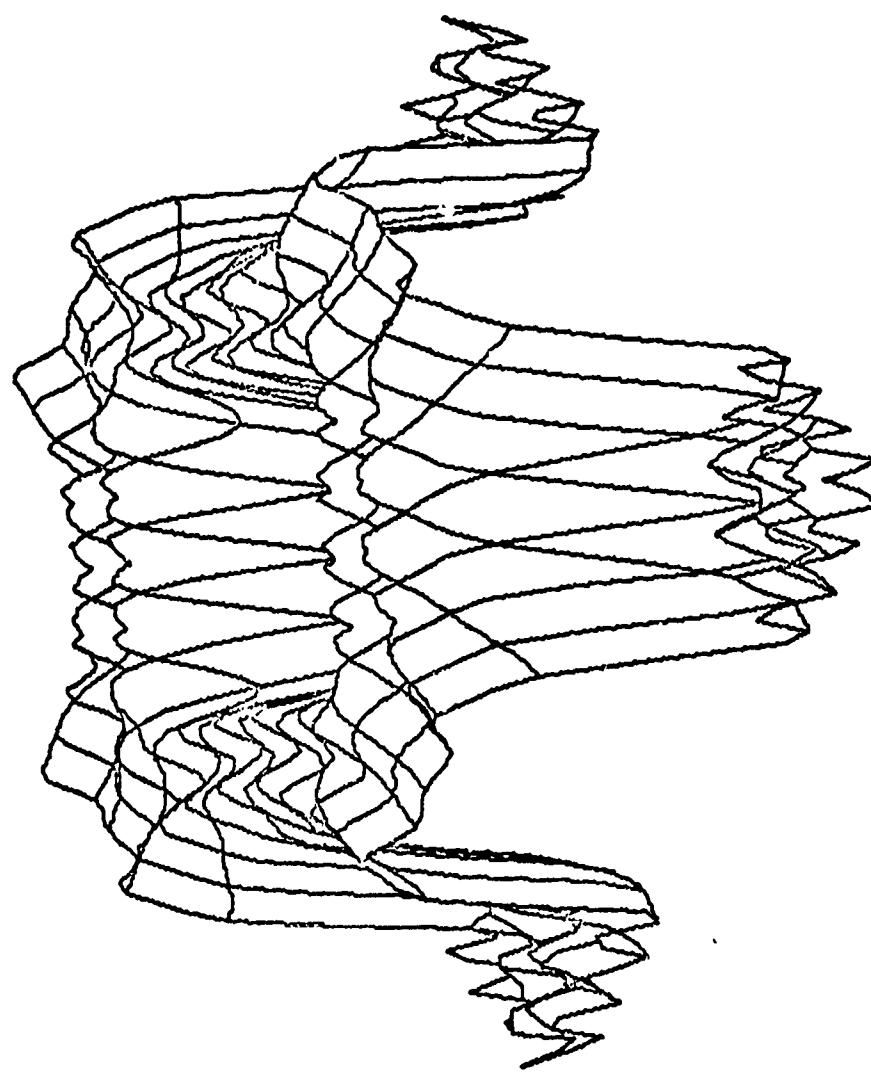


Fig 23. Filter Design 5: Two-Dimensional Frequency Response

VII Conclusions and Recommendations

Conclusions

The transformation of variables technique is a fast and flexible method for designing a class of two-dimensional digital filters. The two-dimensional digital filter design program developed in this investigation extends the work of McClellan and others by providing the user with a systematic method of designing two-dimensional, linear phase, digital filters with a wide variety of shapes. The program is flexible. It can design two-dimensional filters with complex shapes. The program is fast. The design time for a typical 113X113 samples points two-dimensional filter is approximately twelve seconds. The design times of this program are extremely favorable when compared with other available methods.

Although the filter design program is very flexible, certain limitations should be kept in mind. First, the design program maps a specific one-dimensional frequency to an approximation of a desired contour. This approximation may be good or bad depending on how close the shape of the approximation is to the shape of the desired contour. Second, the magnitude characteristic of the frequency response of any two-dimensional digital filter designed by the design program will have quadrantal symmetry. Third, if the first order McClellan Transformation is used in the filter design process, the shapes of the desired two-dimensional

contours must be monotonic. Fourth, if shifting and scaling is required, the results may not be acceptable.

Recommendations for Future Work

There are several things that can be done to improve the performance of the design program developed in this investigation.

1. The order of the one-dimensional prototype filter that is to be designed is an input to program PROTOTYPE. Execution time could be decreased, especially for an inexperienced user, if the program calculated an estimate of the required filter order for the user. Algorithms and formulas that can estimate the required filter order are available in the literature. References 9 through 11 describe a number of these methods.
2. Program PROTOTYPE produces a one-dimensional prototype digital filter with equal ripple filter bands. Since the main output produced by program PROTOTYPE is the impulse response of the one-dimensional filter, other one-dimensional filter design programs could be incorporated into PROTOTYPE. This would give the program tremendous flexibility since filters with other characteristics (monotone passband response, constrained ripple, maximally flat bands, etc.) could also be designed. References 12 through 14 describe programs that could be added to PROTOTYPE.
3. It would be desirable to be able to approximate the

the shape of two or more contours in the the w_2, w_1 plane simultaneously according to some error criteria. This would make the design of multi-band filters much easier.

Bibliography

1. Mersereau, Russell M. "The Design of Arbitrary 2-D Zero-Phase FIR Filters Using Transformations," IEEE Transactions on Circuits and Systems, CAS-27: 142 (February 1980).
2. Mersereau, Russell M., et al. "McClellan Transformations for Two-Dimensional Digital Filtering: I - Design," IEEE Transactions on Circuits and Systems, CAS-23: 405-414 (July 1976).
3. McClellan, James H. "The Design of Two-Dimensional Digital Filters by Transformations," Proceedings of the Seventh Annual Princeton Conference on Information Science and Systems. 247-251 (March 1973).
4. Quatieri, Thomas F. The Design of Two-Dimensional Digital Filters by Generalized McClellan Transformations. MS Thesis, Massachusetts Institute of Technology, June 1975.
5. Stark, Peter A. Introduction to Numerical Methods. London: Macmillan Co., 1970.
6. Oppenheim, Alan V. and Ronald W. Schafer. Digital Signal Processing. Englewood Cliffs: Prentice-Hall, Inc., 1975.
7. Rabiner, Lawrence R. and Bernard Gold. Theory and Application of Digital Signal Processing. Englewood Cliffs: Prentice-Hall, Inc., 1975.
8. McClellan, James H., et al. "A Computer Program for Designing Optimum FIR Linear Phase Digital Filters," IEEE Transactions on Audio and Electroacoustics, AU-21: 506-525 (December 1973).
9. Mintzer, Fred and Bede Liu. "Practical Design Rules for Optimum FIR Bandpass Digital Filters," IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-27: 204-206 (April 1979).
10. Rabiner, Lawrence R. "Approximate Design Relationships for Low-Pass FIR Digital Filters," IEEE Transactions on Audio and Electroacoustics, AU-21: 436-450 (October 1973).
11. Rabiner, Lawrence R. "Some Considerations in the Design of Multiband Finite Impulse Response Digital Filters," IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-22: 462-472 (December 1974).

12. Steiglitz, Kenneth. "Optimal Design of FIR Digital Filters with Nonotone Passband Response," IEEE Transactions on Acoustics, Speech, and Signal Processing, ASSP-27: 643-648 (December 1979).
13. McCallig, T. and Benjamin J. Leon. "Constrained Ripple Design of FIR Digital Filters," IEEE Transactions on Circuits and Systems, CAS-25: 893-901 (November 1978).
14. Edited by the Digital Signal Processing Committee (IEEE). Programs for Digital Signal Processing. USA: IEEE Press, 1979.
15. Brown, Richard B. Investigation of Optimal Linear Shift-Invariant Two-Dimensional Digital Filters. MS Thesis, Air Force Institute of Technology, December 1977.
16. Huang, S. T. "Two-Dimensional Windows," IEEE Transactions on Audio and Electroacoustics, AU-20: 38-89 (March 1972).
17. Hu, J. V. and L. R. Rabiner. "Design Techniques for Two-Dimensional Digital Filters," IEEE Transactions on Audio and Electroacoustics, AU-20: 249-257 (October 1972).

Appendix A

User's Guide for the Two-Dimensional Digital Filter Design Program

This appendix is the user's guide for the filter design program developed in this investigation. It is written so that it can be used as a stand-alone manual for the two-dimensional digital filter design program. The guide is organized as follows. First the general structure of the two-dimensional digital filter design package is described. Then each of the programs and subprograms that make up the package are discussed from the user's point of view. Finally a sample two-dimensional filter design is presented.

Contents

Filter Design Package Structure	82
The Two-Dimensional Digital Filter Design Program	82
Purpose	82
Interactive Nature	84
Filter Size	84
Inputs and Outputs	84
Programs	85
Program CONTROL	85
Program CURFIT	86
Program PROTOTYPE	88
Program GRAPH	90
Subroutine YOURSUB	91
Arguments	91
Example	92
Placement	93
Program PROFILE	93
Procedure COMPILE	95
Procedure DESIGN	95
Procedure ROUTE	95
Procedure PLFILE	95
Procedure 3DPLOT	95
How To Use PROFILE	96
Program PLT3D	98
PLT3D Program Listing	98
Design Example	99
Bibliography	113

Filter Design Package Structure

The two-dimensional digital filter design package consists of the two-dimensional digital filter design program itself, the subroutine YOURSUB, the program PROFILE, and the program PLT3D. All programs except the first one are optional. However, use of the optional programs makes use of the two-dimensional digital filter design program much easier.

All programs are written in Fortran IV extended except program PROFILE which is written in Cyber Control Language. In the discussion that follows, each program will be described from the user's point of view. It will be assumed that the design package is run using the CDC 6600's time sharing system, Intercom.

The Two-Dimensional Digital Filter Design Program

The two-dimensional digital filter design program is written in Fortran IV extended and uses overlays to reduce memory requirements. The program consists of seven separate programs, each resident in a separate overlay. Labeled common storage areas are used to transmit variables between overlays. The program was designed to be run interactively.

Purpose. The two-dimensional digital filter design program designs linear phase, finite impulse response, linear shift-invariant, two-dimensional digital filters. This is done by transforming a one-dimensional digital filter with a frequency response of the form (Ref 1:35-37)

$$H(w) = e^{-jw \cdot k^2} \sum_{n=0}^{k^2} H1D(n) \cos(wn) \quad (112)$$

into a two-dimensional digital filter with a frequency response of the form (Ref 1:33-34)

$$H(w_2, w_1) = e^{-jk^1(w_2 + w_1)} \left[\sum_{l=0}^{k^1} \sum_{m=0}^{k^1} H2D(l, m) \cos(w_1 l) \cdot \cos(w_2 m) \right] \quad (105)$$

by using the generalized McClellan Transformation (ref 2)

$$m(w_2, w_1) = \sum_{i=0}^a \sum_{g=0}^b c(i, g) \cos(iw_1) \cos(gw_2) \quad (3)$$

The filter designer can elect to use either the first order McClellan Transformation ($a=b=1$) or the second order McClellan Transformation ($a=b=2$). The Transformation maps each frequency of the one-dimensional filter to a contour in the two-dimensional frequency plane. The magnitude at each frequency of the one-dimensional frequency response becomes the magnitude along the corresponding contour of the two-dimensional frequency response (Ref 1:10-11).

The program can design lowpass, highpass, bandpass, bandstop, all-pass, and multiband filters. However, only one desired contour in the two-dimensional frequency plane can be closely approximated by the design program. Therefore the program lends itself to the design of lowpass, highpass, and all-pass filters. All filters designed by this program have quadrantal symmetry. This means that the magnitude versus frequency characteristic of the two-dimensional filters

will be four quadrant symmetric.

Interactive Nature. This program is intended for interactive use. This is necessary for two reasons. First, there is no guaranty that the McClellan Transformation will produce a well-defined mapping from the one-dimensional frequency axis to the two-dimensional frequency plane (Ref 1:17-20). If the mapping is ill-defined, filter designer intervention is necessary. Second, the one-dimensional filter order necessary to design a one-dimensional filter with desired characteristics (band error, etc.) is not known in advance. It may be necessary to re-design the one-dimensional filter several times in order to produce the desired filter characteristics.

Filter Size. The program as presently dimensioned can design a two-dimensional digital filter with up to 113 X 113 impulse response samples. The size of the two-dimensional digital filter that can be designed can be changed by re-dimensioning arrays and variables as indicated in the comments of the program listing. Local operating procedures limit interactive programs to 65,000₈ words of central memory. The present array dimensions cannot be increased without exceeding this limit.

Inputs and Outputs. The required inputs to the design program are a set of points that define the contour in the two-dimensional frequency plane that is to be approximated, and the parameters necessary to design the one-dimensional filter that will be transformed into the two-dimensional

filter. The design program outputs the contour approximating function, the impulse response samples of the one-dimensional filter, a graph of the magnitude of the two-dimensional frequency response in the first quadrant (the frequency response has quadrantal symmetry), and the unique two-dimensional impulse response samples. As options, the program can output sets of points that define any two-dimensional contours, a Calcomp plot of the one-dimensional filter's magnitude versus frequency characteristic, a Calcomp plot of the two-dimensional contours, and a Calcomp plot of the two-dimensional filter's magnitude versus frequency characteristic.

Programs. The two-dimensional digital filter design program is composed of seven separate programs. They are called CONTROL, CURFIT, PROTOTYPE, FORCHEB, EXPAND, BACKCHEB, and GRAPH. The design program expects all frequencies to be entered as radians between 0.0 and 1.0. The program automatically multiplies each frequency by pi. In response to no/yes questions, the program expects the user to enter 0 for no and 1 for yes. The important interactive questions and imperative statements will be described program by program. They will be enclosed within quotation marks.

Program CONTROL.

1. "Do you wish to continue?" This question is asked immediately after the one-dimensional prototype filter has been designed. It allows the user to stop execution of the design program at this point and thus end up with only a

one-dimensional filter design.

2. "Do you wish to calculate the 2-D impulse response?"

The user can elect not to generate the two-dimensional impulse response samples. Doing this does not prevent the user from receiving all other output that can be generated by the two-dimensional digital filter design program.

Program CURFIT.

1. "Do you wish to approximate your 2-D contour with a four or nine term approximating function? Enter 4 or 9" If the user enters 4, he is choosing to use the first order McClellan Transformation in the design process. In this case, the contour in the two-dimensional frequency plane that the designer wishes the program to approximate must be monotonic (Ref 1:11-13). Also the resulting two-dimensional filter will have $(NFILET)^2$ impulse response samples (Ref 3). NFILET is the order of the one-dimensional filter that will be transformed into the two-dimensional filter. If the user enters 9, he is choosing to use the second order McClellan Transformation in the design process. In this case, the contour in the two-dimensional frequency plane that the designer wishes the program to approximate does not have to be monotonic. Also the resulting two-dimensional filter will have $(2 \cdot NFILET - 1)^2$ impulse response samples. This is approximately $4 \cdot NFILET^2$.

2. "Do you have a predetermined set of approximating function constants?" If the user wishes to design a two-dimensional filter with the same shape that he used in a previous design, the constants can be entered directly instead

of being re-calculated by the program.

3. "Do you have a predetermined set of constraint equations?" If the user does not wish to use one of the three sets of constraint equations coded into the program, then he must enter his own set of constraint equations. The three sets of constraints coded into the program are the following:

- a. $w = 0$ maps to $(w_2, w_1) = (0,0)$
- b. $w = \pi$ maps to $(w_2, w_1) = (\pi, \pi)$
- c. Both a and b above

4. "Do you wish to enter a set of points that defines your contour?" If the user has not included the subroutine YOURSUB in the code of the two-dimensional digital filter design program or compiled and libraried it using procedure COMPILE of program PROFILE, then he must enter a set of points that defines the desired contour in the two-dimensional frequency plane.

5. "How many points do you wish to enter?" If the user wishes to define the contour by a set of points, he can enter up to 500 points.

6. "If you have a case number enter it; otherwise enter 0." Here the user is going to define the desired contour in the two-dimensional frequency plane by using the subroutine YOURSUB. If the user coded provisions for more than one contour in subroutine YOURSUB, he must enter the value of the variable NCASE that selects the contour that the user wishes to use in the current two-dimensional filter design.

7. "Your contour with your frequency produces an ill-

defined mapping from the w -axis to the w_2, w_1 plane. Enter one of the following option numbers.

- 1) choose different program generated constraints
- 2) enter your own constraint equations
- 3) enter a set of approximating function constants
- 4) start over
- 5) try scaling
- 6) terminate this program"

If the mapping is ill-defined, the filter design process has failed. If the user wants to continue the original filter design, he must either choose one of the other program coded constraints, enter his own constraints, or try scaling.

Scaling produces a well-defined mapping in most cases but the results of the scaling may not be satisfactory from a filter design point of view. Shifting and scaling changes the location of the original specified contour in the two-dimensional frequency plane. The new location may or may not be close to the original location.

8. "Your original frequency has been scaled to "value of the scaled frequency". Enter one of the following option numbers.

- 1) start over
- 2) terminate this program
- 3) continue"

If scaling was used, the filter designer can proceed or stop the design process depending on the result of the scaling.

Program PROTOTYPE. As an aid to understanding the interactive questions in program PROTOTYPE see figure 24.

1. "Enter an odd filter order of "A" or less." If the first order McClellan Transformation was chosen earlier, A equals 113. The filter order must be greater than or equal to

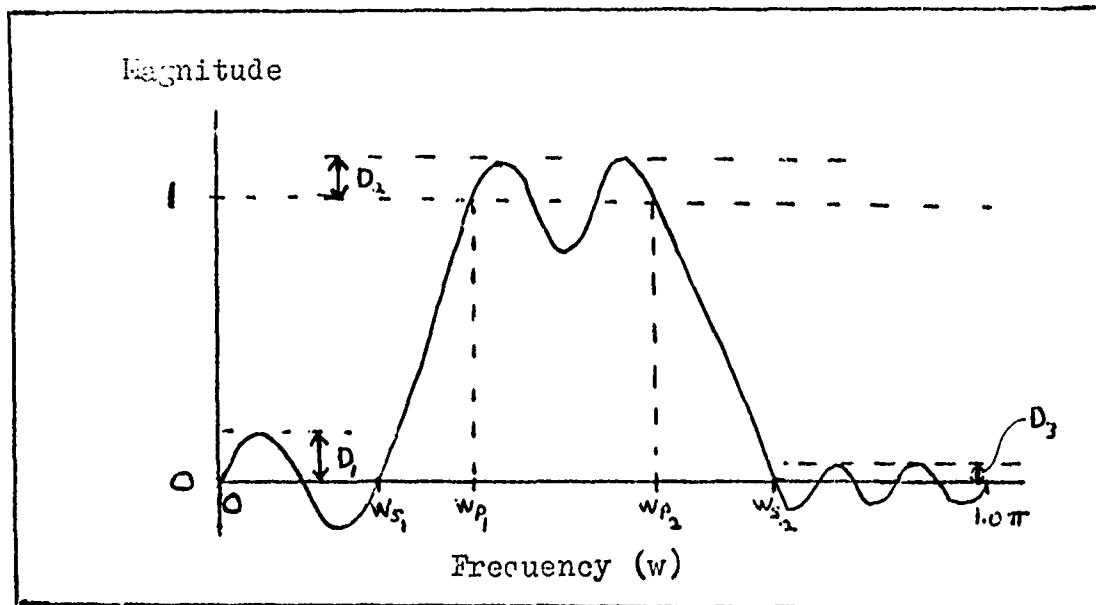


Fig 24. Typical Three Band Filter

three. If the second order McClellan Transformation was chosen earlier, A equals 57. The filter order must still be greater than or equal to three.

2. "How many transition bands does the filter have?" The program can design a one-dimensional filter with up to nine transition bands (10 bands). Figure 24 has two transition bands (3 bands).

3. "Enter the band edge frequencies for each transition band." Here the user enters each w_s and w_p (see figure 24) going from left to right. Each band must be separated by a finite width transition band. There will be one w_s and one w_p for each transition band.

4. "Enter an ideal absolute magnitude for each band of the prototype filter (usually 1 or 0)." Even though program PROTOTYPE produces a filter that is equal ripple in each band,

the ideal magnitude for a passband is 1.0 and the ideal magnitude for a stop band is 0.0.

5. "Enter the ratio of the band errors (one number for each band)." The order of the one-dimensional filter controls the magnitude of the ripple in each band. The user must enter a ratio of the band errors and hope that the resulting band deviations meet his error criteria. If they do not, the one-dimensional filter can be redesigned using a different value for the filter order. The larger the number entered, the smaller the error in the band. For example, if the error ratio for a two-band filter is 10:1; then the error in band 1 will be 10 times smaller than the error in band 2. The deviations output by the program are the D's in figure 24.

6. "Do you wish to redesign the prototype filter?" If the deviations of the designed one-dimensional filter do not meet the user's criteria for band errors, the filter can be redesigned.

Program GRAPH.

1. "Do you want to create a plot file for the Calcomp plotter?" The user is given the option of generating the Calcomp plot of the one-dimensional filter's magnitude versus frequency characteristic and the Calcomp plot of the two-dimensional contours.

2. "Do you want to create a DISSPLA 3-D plot of the frequency response?" The user is given the option of creating a file called tape2. The file tape2 is used by program FIL3D to generate a Calcomp plot of the two-dimensional

filter's magnitude versus frequency characteristic.

Subroutine YOURSUB

Subroutine YOURSUB is used to define the contour in the square region $[0, \pi] \times [0, \pi]$ of the two-dimensional frequency plane that is to be approximated by the two-dimensional digital filter design program. Subroutine YOURSUB generates the points that define the contour. The horizontal axis coordinate of each point is stored in the array W2 and the vertical axis coordinate of each point is stored in the array W1. A maximum of 500 points can be generated by the subroutine. Subroutine YOURSUB must have the following form:

```
SUBROUTINE YOURSUB(NCASE, NPOINTS, W2, W1)
DIMENSION W2(500), W1(500)
```

```
    . . .
    . . .
    . . .
```

User's code that generates the points that define one or more contours in the two-dimensional frequency plane.

```
    . . .
    . . .
    . . .
```

```
RETURN
END
```

Arguments. NCASE allows the user the option of defining more than one contour in the subroutine. The two-dimensional digital filter design program can only approximate one contour per design. NCASE allows the user to designate which of several contours included in the subroutine will be used by the two-dimensional digital filter design program during the design process. NCASE can be used as the variable in a series of if-then statements or in a case statement.

NPOINTS is the number of points that the user's code generates. NPOINTS must be 500 or less. In general the more points generated, the better the approximation to the contour.

W2 is the array in which the horizontal-axis coordinates of each point are stored. W1 is the array in which the vertical-axis coordinates of each point are stored. Both of these arrays are 500 elements long.

Example. The following is a sample subroutine YOURSUB.

```
SUBROUTINE YOURSUB (NCASE, NPOINTS, W2, W1)
DIMENSION W2(500), W1(500)
```

```
IF (NCASE. NE. 1) GO TO 20
NPOINTS = 41
```

```
A = .3
```

```
XMIN = .5*A**2
```

```
DO 10 I=1,41
```

```
W2(I) = ((1.0 - XMIN)*(I - 1.0)/40 + XMIN)
```

```
W1(I) = XMIN/W2(I)
```

```
10      CONTINUE
      RETURN
```

```
20      NPOINTS = 100
```

```
R = .7
```

```
DO 100 LZ=1,100
```

```
W2(LZ) = (LZ - 1)/99 * R
```

```
W1(LZ) = SQRT(R**2 - W2(LZ)**2)
```

```
100     CONTINUE
```

```
      RETURN
```

```
END
```

If, during the execution of the two-dimensional digital filter design program, the user enters NCASE = 1; the program will approximate a hyperbola with equation $w_1 = a^2/(2w_2)$. Forty-one points will be used to define this hyperbola. If, on the other hand, the user enters NCASE = 2; the program will approximate a quarter circle centered at the origin with radius r and equation $w_1 = (r^2 - w_2^2)^{1/2}$.

One-hundred points will be used to define this quarter circle.

Placement. Subroutine YOURSUB can be inserted into the two-dimensional digital filter design program deck in the back of the program called CURFIT. It can also be written in the create mode of editor, saved as a local file called X, and then compiled and libraried by the program PROFILE.

Program PROFILE

Program PROFILE is used to process files needed by or created by the two-dimensional digital filter design program and program PLT3D. PROFILE is written in Cyber Control Language. The program consists of five separate procedures. Although this program is optional, equivalent commands must be given in order to process the files needed by or created by the two-dimensional digital filter design program and program PLT3D if program PROFILE is not used. To use PROFILE the user must do the following:

1. Store program PROFILE in a permanent file called Profile.
2. Store the binary version of the two-dimensional digital filter design program in cycle 490 of a permanent file called Mcltrn.
3. Store program PLT3D (only if used) in cycle 490 of a permanent file called D3PLOT.

A listing of program PROFILE and a discussion of the five procedures (CCPILE, DESIGN, ROUTE, FILE, 3DILCT) will be presented next.

PROFILE Program Listing.

```
.PPOC,COMPILE,TERM=.
FTP,1=NAME,LIB=LIBRARY.
I=4,FILE=NAME,LIB=LIBRARY.
RETURN,NAME,LIB.
ENDIF,JUMP.
SMITLIB,I=14,LIB=LIB.
RETURN,SCSC,LIB=LIB.
.DATADB
LIBRARY,MYLIB,NAME.
ADD♦,6000.
FINISH.
ENDRUN.
♦EOF
.PPOC,DESIGN,NAME=.
RETURN,NAME.
ATTACH,CIC,CCPLDT56%,ID=LIBRARY,DA=ADD.
LIBRARY,MYLIB,CIC.
ATTACH,MX,MOLTAH,CY=490.
MX.
RETURN,MX,MOL,CIC.
LIBRARY.
♦EOF
.PPOC,ROUTE,TERM=BB,USER=CIC.
REWIND,RESULT,PLOT.
REQUEST,PRX,♦0.
COPYSBF,RESULT,PRX.
ROUTE,PRM,TID=TERM,FID=USER,DC=PR,ST=CSB.
RETURN,RESULT,PRX.
REQUEST,CEC,♦0.
COPY,PLOT,CEC.
ROUTE,CEC,DC=PT,TID=TERM,FID=USER,ST=CSB.
RETURN,PLOT,CEC.
♦EOF
.PPOC,3DPLT,TERM=BB,USER=CIC.
ATTACH,PLFILE,YOURFILE.
ATTACH,DISSPLA,ID=LIBRARY,SM=ASD.
LIBRARY,DISSPLA.
ONLINE.
REWIND,PLOT.
REQUEST,DAV,♦0.
COPY,PLOT,DAV.
ROUTE,DAV,DC=PT,TID=TERM,FID=USER,ST=CSB.
RETURN,DISSPLA,PLOT,PLFILE.
LIBRARY.
♦EOF
♦EOF
```

Procedure COMPILE. This procedure compiles and libraries the subroutine YOURSUB. The procedure assumes that subroutine YOURSUB has been saved without sequence numbers in a local file called X.

Procedure DESIGN. This procedure first attaches the Calcomp plotter routines (file CCPLOT56X). It then declares the plotter file and the file containing subroutine YOURSUB (YLIB) to be libraries. Finally it begins execution of the two-dimensional digital filter design program (binary assumed to be stored in cycle 490 of file McLtrn).

Procedure ROUTE. This procedure sends the hard copy output file (RESULT) to the printer and the Calcomp plotter file (PLOT) to the plotter. The hardcopy output file will have the banner CIC. The banner can be changed to any banner the user desires by changing USER=CIC in the first statement of the procedure to USER="banner the user desires". The plot generated by the Calcomp plotter will always have as the banner, the ID that the computer system assigns the user just after he logs on to the Intercom system. The user has no control over this banner.

Procedure PLFILE. This procedure first saves the information on the file called TAPE2 in a permanent file called DATA3DPLOT. It then batches program FIL3D (assumed to be stored in cycle 490 of a file called D3PLOT) to the input queue. Program FIL3D will purge file DATA3DPLOT after it finishes executing.

Procedure 3DPLOT. This procedure first attaches a

permanent file called YOURFILE (created by program PLT3D). It then attaches the file DISPLA containing the three-dimensional figure plotting routines and starts execution of these routines. Finally it sends the file generated by the three-dimensional figure plotting routines (PLOT) to the Calcomp pictter. Again the banner on the plot will be the ID assigned the user after the user has logged on the Intercom system. The file YOURFILE will exist for eight days. During this time procedure 3DPLOT can be re-executed to get multiple copies of the three-dimensional plot.

How to Use PROFILE. Typically the following sequence of actions and commands would occur during the design of a two-dimensional digital filter.

1. The user writes the subroutine YOURSUB in the create mode of editor. Subroutine YOURSUB is saved as local file X without sequence numbers (i.e. SAVE, X, NOS).
2. The command "B,B" is given.
3. The command "ATTACH, PROFILE" is given.
4. The command "BEGIN, COMPILE, PROFILE" is given. If the subroutine YOURSUB compiles with errors, the user must correct the mistakes by using the edit mode of editor. Then the corrected subroutine is saved as local file X over-writting the old file X (i.e. SAVE, X, OVER, NOS). Steps 2 and 4 are then repeated.
5. The command "BEGIN, DESIGN, PROFILE" is given.
6. The command "BEGIN, ROUTE, PROFILE" is given.
7. The command "BEGIN, RFILE, PROFILE" is given. The

terminal will respond with a list of files. The user should write down or remember the file listed under remote input files.

8. The user must now wait until the program PLT3D finishes execution. When the user gives the command "FILES" and the file that was listed under remote input files is now listed under remote output files, then PLT3D has finished executing.

9. The command "BATCH, "remote output file name", LOCAL" is given.

10. The command "EDITOR" is given.

11. The command "EDIT, "remote output file name", S" is given.

12. The command "L, A" is given. The user must check the program listing for errors. The most common error is no file space available for the creation of the file YOURFILE. If errors have occurred, the user must go back to step 7 and proceed from there.

13. The command "BEGIN, 3DPLOT, PROFILE" is given. The terminal responds with DISSPLA POSTPROCESSOR FOR ONLINE CALCOMP PLOTTER. ENTER DIRECTIVES. The user enters the command "DRAW-1-END". When the terminal responds with END ONLINE, the sequence is complete.

It should be noted that if the subroutine YOURSUB is inserted into the two-dimensional digital filter design program deck or if a set of points is going to be input into the two-dimensional digital filter design program during execution,

then the user skips steps 1, 2, and 4.

Program PLT3D

Program PLT3D generates the magnitude of the two-dimensional frequency response at a grid of points on the two-dimensional frequency plane. The coordinates of each point on the grid and the magnitude at each point are stored on a file called YOURFILE. The program uses the information stored on file DATA3DPLOT (originally TAPE2). If the program runs successfully, the file DATA3DPLOT is purged and the file YOURFILE is created.

To use this program, the user must replace the current job card with his own job card. A listing of the program is given below.

PLT3D Program Listing.

```
CI4,CM12: 1 .T700303,0700P0L0LLA,4162
FTN(R=.)
ATTACH,TAPE2,DATA3DPLOT.
ATTACH,BIGSFLA,,1=L12111,3N=150.
LIBRARY,PISSFLA.
REQUEST,PLFILE,,PF.
LGO.
CATALOG,PLFILE,YOURFILE.
PURGE,TAPE2.
      PROGRAM PLT3D(THINUT,OUTPUT,TAPE2,PLFILE=0)
      DIMENSION WORK(500),H10CHFB(50),CURVFIT(9)
      COMMON /C15/ H10CHFB, CURVFIT, K0
      EXTERNAL ;
      P50D (2,10) K1
10:  FORMAT(13)
      RE1D (2,20) (CURVFIT('R'),MR=1,0)
20:  FORMAT(1E18.9,/,1E18.9)
      REA7(2,3L) (H10CHFB(4T),4T=1,K0)
30:  FORMAT(9(7E11.0,/,))
      CALL COMERS
      CALL BGNFL(1)
      CALL TITL3D(14,-1,7,4,4)
      CALL VU47(12.0,0.0,0.0)
      CALL AXLES3D(1,1,0,1,1,1,2.0,2.0,1.0)
```

```

CALL CRAFT(-1.0, 1.0, -1.0, 1.0, -0.25, 0.1, 1.25)
CALL CURFUN(G, 3.125, 7.125, WORK)
CALL ENDFL(1)
CALL ENDFL
STOP V = J
FUNCTION G(V)
COMMON /CIE/ M7000, C11 VFIT, K0
DIM M7000 /7048(7), CURVFIT(7)
PI=3.14159265
W2 = V * PI
W1 = V * PI
G = M10*W2/1
DO 1, M=2,K0
1 G = G + CURVFIT(M)*((CURVFIT(1) + CURVFIT(3)*COS(W2)
+ CURVFIT(5)*COS(2*W2)+CURVFIT(2)*COS(W1)
+ CURVFIT(4)*COS(W1)+COS(W2)
+ CURVFIT(6)*COS(W1)*COS(2*W2)
+ CURVFIT(7)*COS(2*W1)
+ CURVFIT(8)*COS(2*W1)+COS(W2))** (M-1))
CONTINUE
15' RETURN $ -NO

```

Design Example

The following example illustrates the use of PROFILE and the two-dimensional digital filter design program. The desired contour is a quarter circle centered at the origin with radius $.7\pi$ and equation $w_1 = (r^2 - w_2^2)^{\frac{1}{2}}$. Note that user entries are underlined.

11-24-80 LOGGED IN AT 18.12.39.
WITH USER-ID P9
EQUIP/PORT 14/066

COMMAND- EDITOR

..CREATE

```
100= SUBROUTINE YOUSUB(NCASE,NPOINTS,M2,M1)
110= DIMENSION M2(500),M1(500)
120= NPOINTS = 100
130= R = .7
140= DO 100 LZ=1,100
150= M2(LZ) = (LZ-1)*99 + R
160= M1(LZ) = SQRT(R**2 - M2(LZ)**2)
170= 100 CONTINUE
180= RETURN
190= END
200==
```

..L,A

```
100= SUBROUTINE YOUSUB(NCASE,NPOINTS,M2,M1)
110= DIMENSION M2(500),M1(500)
120= NPOINTS = 100
130= R = .7
140= DO 100 LZ=1,100
150= M2(LZ) = (LZ-1)*99 + R
160= M1(LZ) = SQRT(R**2 - M2(LZ)**2)
170= 100 CONTINUE
180= RETURN
190= END
```

..SAVE,X,MOS

..E,B

COMMAND- ATTACH,PROFILE

PFN 12

PROFILE:

AT CY= 500 SH=INIT

COMMAND- BEGIN,COMPILE,PROFILE

.035 CP SECOND COMPILE TIME

COMMAND- BEGIN,DESIGN,PROFILE

AT CY= 999 SH=END

PROGRAM FOR THE DESIGN OF TWO-DIMENSIONAL FINITE IMPULSE RESPONSE
FILTERS USING THE "M" TRANSFORMATION

ENTER ALL FREQUENCIES IN RADIANS BETWEEN 0.0 AND 1.0. THE PROGRAM
WILL AUTOMATICALLY MULTIPLY TIMES PI.

IN RESPONSE TO YES/NO QUESTIONS: ENTER 0 FOR NO AND 1 FOR YES.

DO YOU WISH TO APPROXIMATE YOUR 2-D CONTOUR WITH A FOUR OR NINE TERM APPROXIMATING FUNCTION? ENTER 4 OR 9.

4

DO YOU HAVE A PREDETERMINED SET OF APPROXIMATING FUNCTION CONSTANTS?

0

ENTER THE 1-D FREQUENCY THAT YOU WISH TO MAP TO YOUR 2-D CONTOUR (0.0 TO 1.0 <X PI>).

.3

DO YOU HAVE A PREDETERMINED SET OF CONSTRAINT EQUATIONS?

0

CHOOSE ONE OF THE FOLLOWING CONSTRAINTS BY NUMBER.

- 1) W=0 MAPS TO (W2,W1)=(0,0)
- 2) W=PI MAPS TO (W2,W1)=(PI,PI)
- 3) BOTH 1 AND 2 ABOVE

1

DO YOU WISH TO ENTER A SET OF POINTS THAT DEFINES YOUR CONTOUR?

0

IF YOU HAVE A CASE NUMBER ENTER IT; OTHERWISE ENTER 0.

0

YOUR CONTOUR WITH YOUR FREQUENCY PRODUCES AN ILL-DEFINED MAPPING FROM THE W-AXIS TO THE W2,W1 PLANE. ENTER ONE OF THE FOLLOWING OPTION NUMBERS.

- 1) CHOOSE DIFFERENT PROGRAM GENERATED CONSTRAINTS
- 2) ENTER YOUR OWN CONSTRAINT EQUATIONS
- 3) ENTER A SET OF APPROXIMATING FUNCTION CONSTANTS
- 4) START OVER 5) TRY SCALING 6) TERMINATE THIS PROGRAM

5

YOUR ORIGINAL FREQUENCY HAS BEEN SCALED TO .280320PI. THE SCALED FREQUENCY WILL BE MAPPED TO YOUR CONTOUR.

ENTER ONE OF THE FOLLOWING OPTION NUMBERS.

- 1) START OVER
- 2) TERMINATE THIS PROGRAM
- 3) CONTINUE

3

LINEAR LEAST SQUARES APPROXIMATION WITH ONE TRAJECTORY

THE APPROXIMATION HAS THE FORM: $W_2(W_1) = A + B \cdot W_1 + C \cdot W_1^2$
 $I \cdot W_1 \cdot W_1^2 + C \cdot W_1^3$ WITH

$$A = .14520 \quad B = .50000 \quad C = .50000 \quad D = -.04820$$

CONSTRAINT EQUATIONS

$$1.00 = 1.00A + 1.00B + 1.00C + 1.00D$$

THE 1-D FREQUENCY THAT MAPS TO THE USERS 2-D CONTOUR = .230320PI.

ENTER AN ODD FILTER ORDER OF 113 OR LESS.

61 HOW MANY TRANSITION BANDS DOES THE FILTER HAVE?

1 ENTER THE BAND EDGE FREQUENCIES FOR EACH TRANSITION BAND.

.1,.15 ENTER AN IDEAL ABSOLUTE MAGNITUDE FOR EACH BAND OF THIS PROTOTYPE FILTER (USUALLY 1 OR 0).

1,0 ENTER THE RATIO OF THE BAND ERRORS (ONE NUMBER FOR EACH BAND). FOR EXAMPLE A 3 BAND FILTER MIGHT HAVE AN ERROR RATIO OF 1, 10, 5.

2,1

ONE-DIMENSIONAL FIR PROTOTYPE FILTER DESIGN

FILTER LENGTH = 61

	BAND 1	BAND 2	BAND 3
LOWER BAND EDGE	0.000000000	.150000000	
UPPER BAND EDGE	.100000000	1.000000000	
DESIRED VALUE	1.000000000	0.000000000	
WEIGHTING	2.000000000	1.000000000	
DEVIATION	.017953356	.035906712	
DEVIATION IN DB	-34.917037125	-28.896487212	

DO YOU WISH TO REDESIGN THE PROTOTYPE FILTER?

0

DO YOU WISH TO CONTINUE?

1

TWO-DIMENSIONAL FIR FILTER DESIGN

61 BY 61 SAMPLE POINT FILTER

MAGNITUDE OF THE 2-D FREQUENCY RESPONSE IN THE -180° QUADRANT
(THE FREQUENCY RESPONSE IN 45° QUADRANT IS SAME)

(C) 01-# 01-TIME1 P1

+

+

+

1.0+ .03 -.01 .04 .03 .03 -.02 -.03 .03 -.01 -.02 .04

+

+

+

+

.9+ .04 .00 .03 .03 .04 .01 -.03 -.01 .04 .03 -.01

+

+

+

+

.8+ .02 .03 .00 -.02 -.00 .02 .04 .02 .02 .04 -.01

+

+

+

+

.7+ -.03 .02 -.04 -.02 -.01 -.02 -.03 -.02 .02 -.01 .01

+

+

+

+

.6+ -.00 -.04 .03 .03 .01 -.01 -.03 -.03 .04 -.03 -.01

+

+

+

+

.5+ .02 .03 -.03 -.02 .02 .03 -.01 -.02 .02 .01 -.02

+

+

+

+

.4+ .04 -.04 .03 -.00 -.03 .02 .01 -.01 -.00 .04 .03

+

+

+

+

.3+ .77 .48 -.03 .03 -.00 -.02 .03 -.02 -.02 .03 .03

+

+

+

.2+ 1.00 1.02 .62 -.03 .03 -.03 .03 -.04 .00 .03 .04

+

+

+

+

.1+ 1.01 .99 1.02 .48 -.04 .03 -.04 .02 .03 .00 -.01

+

+

+

0.0+ .98 1.01 1.00 .77 .04 .02 -.00 -.03 .02 .04 .03

+++++

0.0 .1 .2 .3 .4 .5 .6 .7 .8 .9 1.0

W2-AXIS (TIMES P1)

(C) DO YOU WISH TO CALCULATE THE E-D IMPULSE RESPONSE?

1

(THE TWO-DIMENSIONAL IMPULSE RESPONSE IS BEING
WRITTEN TO YOUR HARDCOPY OUTPUT FILE (RESULT)).

DO YOU WISH TO HAVE TABULAR DATA FOR ANY CONTOURS IN THE FIRST
QUADRANT OF THE W2,W1 PLANE PRINTED OUT ON YOUR HARDCOPY OUTPUT FILE
(RESULT)?

1

ENTER W (0.0 TO 1.0) OR ENTER 2 TO CONTINUE WITH THE PROGRAM.

0

HOW MANY POINTS DO YOU WANT OUTPUT?

7

FOR W = 0.0000PI, (W2,W1) =:

W2:	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
W1:	.0000	.0000	.0000	.0000	.0000	.0000	.0000

ENTER W (0.0 TO 1.0) OR ENTER 2 TO CONTINUE WITH THE PROGRAM.

.15

HOW MANY POINTS DO YOU WANT OUTPUT?

7

FOR W = .1500PI, (W2,W1) =:

W2:	0.0000	.0663	.1327	.1990	.2653	.3317	.3980
W1:	.3999	.3925	.3771	.2771	.2112	.1392	.0215

ENTER W (0.0 TO 1.0) OR ENTER 2 TO CONTINUE WITH THE PROGRAM.

.1

HOW MANY POINTS DO YOU WANT OUTPUT?

7

FOR W = .1000PI, (W2,W1) =:

W2:	0.0000	.0427	.0853	.1280	.1707	.2133	.2560
W1:	.2577	.2511	.2323	.2029	.1642	.1144	.0516

ENTER W (0.0 TO 1.0) OR ENTER 2 TO CONTINUE WITH THE PROGRAM.

2

(ANY POINTS WITH COORDINATES (-.15,-.25) ARE INVALID AND SHOULD
BE DISREGARDED)

DO YOU WANT TO CREATE A PLOT FILE FOR THE CALCOMP PLOTTERT?

DO YOU WANT TO CREATE A DISPLAY 3-D PLOT OF THE FREQUENCY RESPONSE?

• THE DATA REQUIRED FOR THE DIFFLA PLOT IS NOW ON TAPE 6.

11128 NORMAL PROGRAM TERMINATION
052100 MAXIMUM EXECUTION FL.
4.640 CP SECONDS EXECUTION TIME.
COMMAND - BEGIN,ROUTE,PROFILE
COMMAND- BEGIN,PLFILE,PROFILE
--LOCAL FILES--
MYLIB \$INPUT \$OUTPUT ♦PROFILE
--REMOTE INPUT FILES--
CI4004.J
NEWCYCLE CATALOG
PP = 008 DAYS
CT ID= T730303 PFM=DATA3DPILOT
CT CY= 003 SN=RFIT 000000256 WORDS.:
COMMAND- FILES
--LOCAL FILES--
MYLIB \$INPUT \$OUTPUT ♦PROFILE
--REMOTE OUTPUT FILES--
CI4004.J
COMMAND- BATCH,CI4004.J,LOCAL
COMMAND- EDITOR
..
YOU HAVE AN EXISTING EDIT FILE
EDIT,CI4004.J.S
LINES TRUNCATED- CH= 72 CHARS, LONGEST L
.L,A

```

100=1           PROGRAM PLT3D      74/74     OPT=1
110=0
120=           1           PROGRAM PLT3D(INPUT,OUTPUT,TAPE2,PLFILE
=0)
130=           DIMENSION WORK(650),H1DCHEB(57),CURVFIT
(9)
140=           COMMON /ONE/ H1DCHEB, CURVFIT, K0
150=           EXTERNAL G
160=           5           READ (2,10) K0
170=           10          FORMAT(I3)
180=           READ (2,20) (CURVFIT(MR),MR=1,9)
190=           20          FORMAT (5E18.9,/,4E18.9)
200=           READ (2,30) (H1DCHEB(MT),MT=1,K0)
210=           10          FORMAT (9(7E18.9,/,))
220=           CALL COMPRS
230=           CALL BGMPL(1)
240=           CALL TITL3D(1H , -1,7.0,6.0)

```

```

250=          15          CALL M4RBL(12,10,12,1)
260=          15          CALL R_E130(0,0,0,0,0,0,0,0,1,5)
270=          15          CALL GRBF30(-1.0+0.1,1.0,-1.0,0.1,1,1,1,1)
280=          15
290=          15
300=          15
310=          20          FUNCTION G
320=          20          74/74  UPT=1
330=          20
340=          1          FUNCTION G(X,Y)
350=          1          COMMON /ONE/ H1DCHEB, CURVFIT, FD
360=          1          DIMENSION H1DCHEB(57), CURVFIT(9)
370=          1          PI=3.1415926536
380=          5          M0 = X * PI
390=          5          M1 = Y * PI
400=          5          G = H1DCHEB(1,
410=          5          DO 150 M=2,FD
420=          5          G = G+ H1DCHEB(M)*CURF
FIT(1)
430=          10         1          + CURVFIT(8)*COS
2*M2+
440=          10         2          + CURVFIT(4)*COS
M12*COS
450=          10         3          + CURVFIT(5)*COS
M12*COS
460=          10         4          + CURVFIT(7)*COS
2*M1)
470=          10         5          + CURVFIT(6)*COS
2*M1)*C
480=          15         6          + CURVFIT(9)*COS
2*M1)*C
490=          150        150  CONTINUE
500=          150        RETURN & END
510=          150
520=          150
530=          150          PLOTTING COMMENCING
540=          150          -----
550=          150
560=          150          .... DISSPLA VERSION 7.2 ....
570=          150          NO. OF FIRST PLOT  1
580=
590=
600=
610=
620=
630=
640=
650=
660=
670=          PLOT NO.  1 WITH THE TITLE
680=
690=          HAS BEEN COMPLETED.

```

700=
710= PLOT ID. PERIOD
720= PLOT 1 18.25.17 MON 24 NOV 1980 JOB=CI4004J . MPHES -
D DIS
730=1
740=
750= .
760= . WORK PC'-DIMENSIONS
770= . -----
780= .
790= . X3DPC13= 2.00
800= . Y3DPC13= 2.00
810= . Z3DAKIT= 1.50
820= . IN AEE. 3-D UNITS
830= .
840=
850= .
860= . VIEWPOINT
870= . -----
880= .
890= . XVU= 1.200E+01
900= . YVU= 1.000E+01
910= . ZVU= 1.200E+01
920= . IN AEE. 3-D UNITS
930= .
940=
950= .
960= . GRAPH SET-UP (GRAF3D)
970= . -----
980= .
990= . ORIGIN
1000= . -----
1010= . X3DORIGIN=-1.000E+00
1020= . Y3DORIGIN=-1.000E+00
1030= . Z3DORIGIN=-2.500E-01
1040= .
1050= . STEP SIZE
1060= . -----
1070= . X3DSTP= 1.000E-01
1080= . Y3DSTP= 1.000E-01
1090= . Z3DSTP= 1.000E-01
1100= .
1110= . MAXIMUM
1120= . -----
1130= . X3DMAX= 1.000E+00
1140= . Y3DMAX= 1.000E+00
1150= . Z3DMAX= 1.250E+00
1160= .
1170=
1180= .
1190=
1200= . LOCATION OF CURRENT PHYSICAL ORIGIN .
1210= . X= 2.25 Y= .82 INCHES .

1220= . FROM LOWER LEFT CORNER OF PAGE .
 1230=
 1240=
 1250=
 1260=
 1270=
 1280=
 1290=
 1300=
 1310= END DISSPLA -- 2332 VECTORS GENERATED IN 1 PLOT FRAMES.
 1320=♦EDR
 1330=1 CSA NDS/BE L518C L518C-CMR1 10/20/80
 1340= 18.25.22.CI4004J FROM /00
 1350= 18.25.22.IP 00000384 WORDS - FILE INPUT , DC 04
 1360= 18.25.22.CI4.CM120000.T790303,CICCOLELLA,4162
 1370= 18.25.23.FTN(R=0)
 1380= 18.25.28. .131 CP SECONDS COMPILATION TIME
 1390= 18.25.28.ATTACH,TAPE2,DATA3DPILOT.
 1400= 18.25.29.AT CY= 003 SN=AFIT
 1410= 18.25.29.ATTACH,DISSPLA, ID=LIBRARY,SN=ASD.
 1420= 18.25.29.PFN IS
 1430= 18.25.29.DISSPLA
 1440= 18.25.29.AT CY= 999 SN=ASD
 1450= 18.25.29 LIBRARY,DISSPLA.
 1460= 18.25.29.REQUEST,PLFILE,♦PF.
 1470= 18.25.31.L60.
 1480= 18.25.38. NON-FATAL LOADER ERRORS -
 1490= 18.25.38.NON-EXISTENT LIBRARY GIVEN - SYSIO
 1500= 18.25.38. NON-FATAL LOADER ERRORS -
 1510= 18.25.38.NON-EXISTENT LIBRARY GIVEN - SYSIO
 1520= 18.26.19. STOP
 1530= 18.26.19. 070300 MAXIMUM EXECUTION FL.
 1540= 18.26.19. 7.330 CP SECONDS EXECUTION TIME.
 1550= 18.26.19.CATALOGS,PLFILE,YOURFILE.
 1560= 18.26.19.MEMOCYCLE CATALOG
 1570= 18.26.19.FP = 008 DAYS
 1580= 18.26.19. CT ID= T790303 PFN=YOURFILE
 1590= 18.26.19. CT CY= 011 SN=AFIT 00000256 WORDS.
 1600= 18.26.19.PURGE,TAPE2.
 1610= 18.26.19.PP ID= T790303 PFN=DATA3DPILOT
 1620= 18.26.19.PP CY= 003 SN=AFIT 00000256 WORDS.
 1630= 18.26.19.OP 00000F40 WORDS - FILE OUTPUT , DC 40
 1640= 18.26.20.MJ 3584 WORDS 14336 MHZ USEIN
 1650= 18.26.20.CPA 9.077 SEC. 6.552 ADJ.
 1660= 18.26.20.IC 10.928 SEC. 7.234 ADJ.
 1670= 18.26.20.CM 392.526 FWD. 1.877 ADJ.
 1680= 18.26.20.CPUS 11.694
 1690= 18.26.20.CCST .77
 1700= 18.26.20.PP 13.150 SEC. DATE 11-24-80
 1710= 18.26.20.EJ END OF JOB. 00 T790303.
 ..B.B
 COMMAND- BEGIN,3DPILOT,PROFILE
 DISCPLA POSITIONED FOR ON-LINE CALCEMP PLOTTER.

ENTER DIRECTIVES.

AT CY= 011 CM=AFIT

PFM IS

DISPLAY:

AT CY= 999 SN=ASD IPAW-1-END\$

THE FOLLOWING PLOTS HAVE BEEN PROCESSED

PLOT 1

END POSTPROCESSOR
END ONLINE

W1-AXIS (X PI)

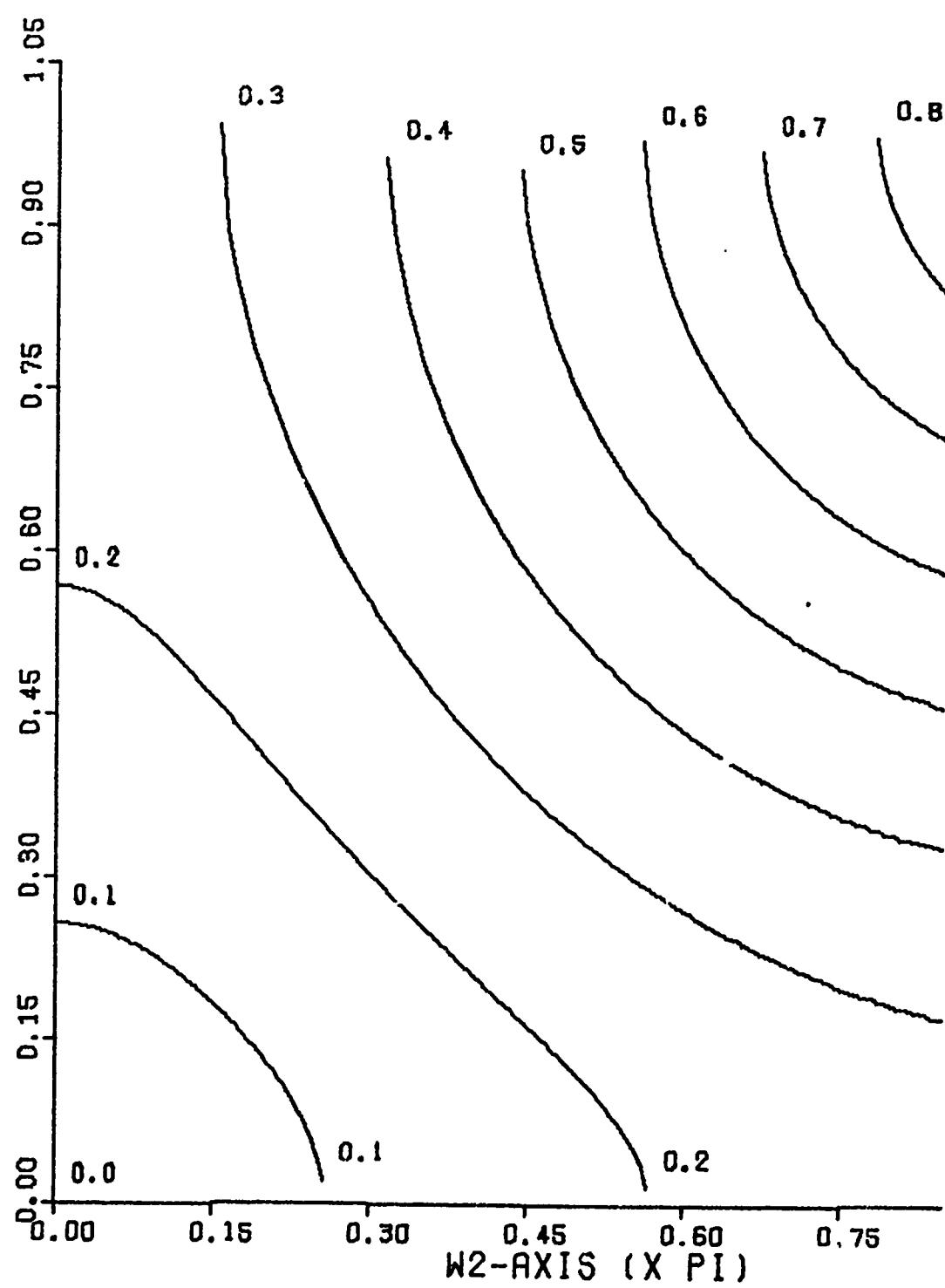


Fig 25. Design Example: Two-Dimensional Contours

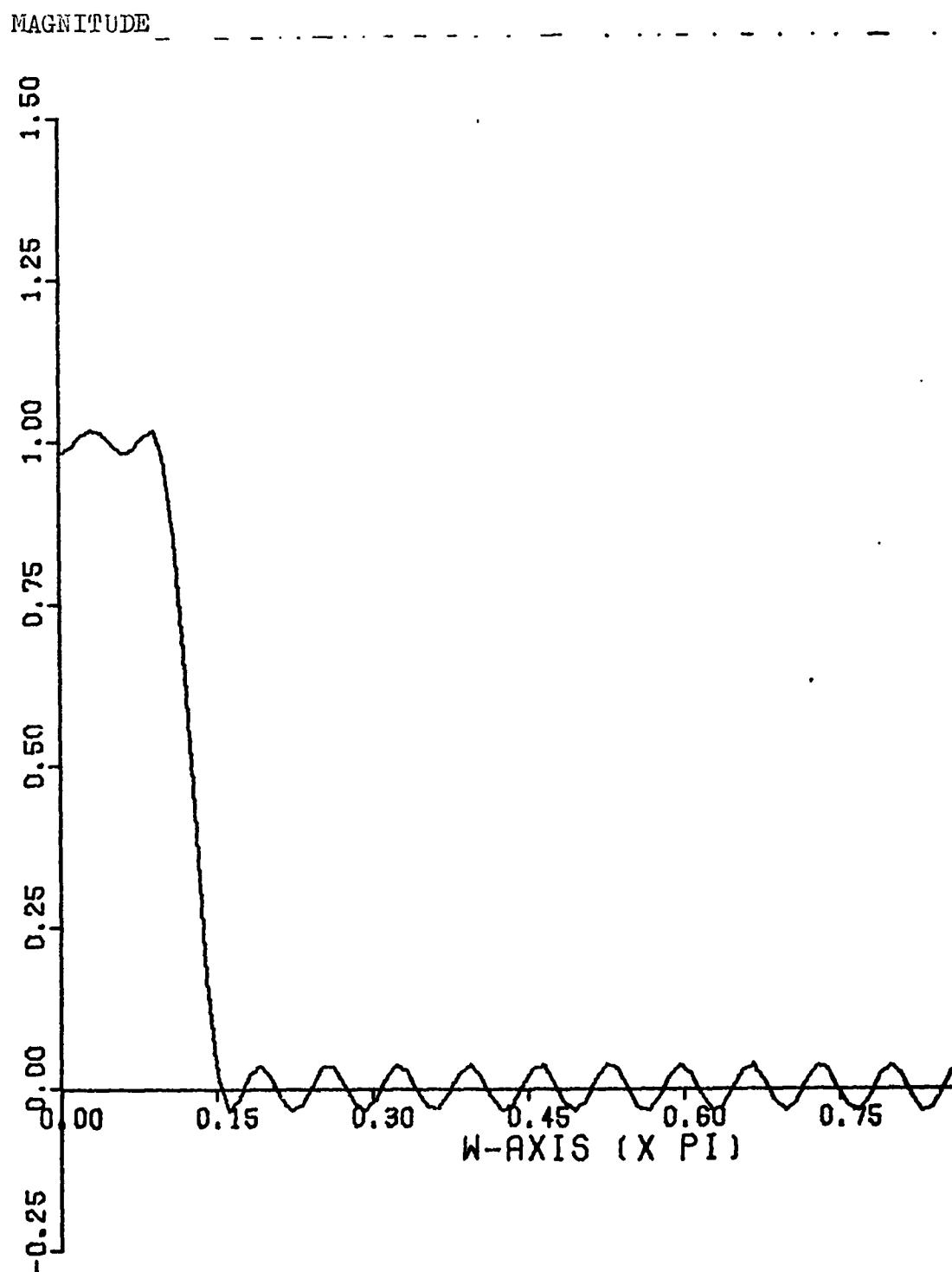


Fig 26. Design Example: One-Dimensional Frequency Response

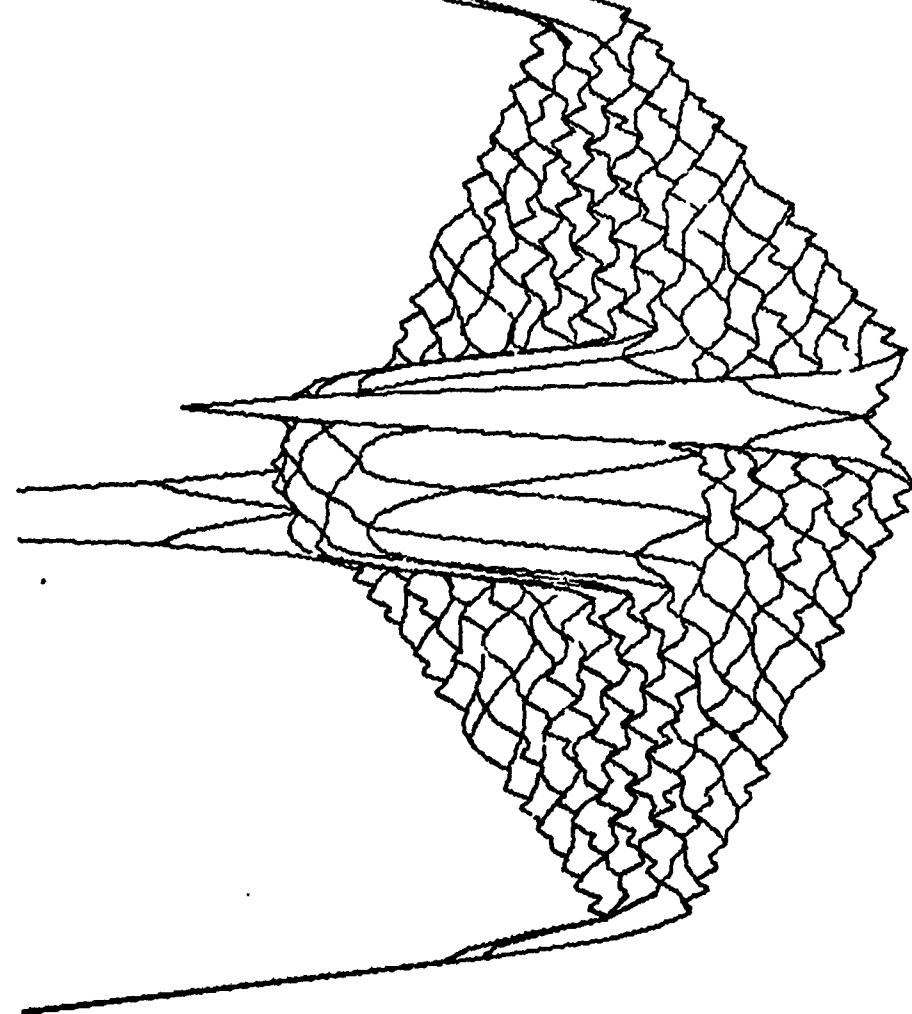


Fig 27. Design Example: Two-Dimensional Frequency Response

Bibliography

1. Ciccolella, David. Design of Linear Phase, Finite Impulse Response, Two-Dimensional, Digital Filters. MS Thesis, Air Force Institute of Technology, December 1980.
2. Mersereau, Russell N., et al. "McClellan Transformations for Two-Dimensional Digital Filtering: I - Design," IEEE Transactions on Circuits and Systems, CAS-23: 405-424 (July 1976).
3. McClellan, James H. "The Design of Two-Dimensional Digital Filters by Transformations," Proceedings of the Seventh Annual Princeton Conference on Information Science and Systems. 247-251 (March 1973).

Appendix B

Listing of the Two-Dimensional Digital Filter Design Program

This appendix contains a listing of the two-dimensional digital filter design program developed in this investigation. The program is written in Fortran IV extended and uses overlays to reduce memory requirements.


```

20 FORMAT (/,4X,39'0 DO YOU WISH TO APPROXIMATE YOUR 2-D CONTOUR WITH
1A FOU OF NINE FROM /,4X,39'0 APPROXIMATING FUNCTION? ENTER 4 OR 9.
2      ,/)
      READ*,MYOPT2
      IF(MYOPT2 .NE. 4 .AND. MYOPT2 .NE. 9) GO TO 1
      PRINT 1
40 FORMAT (/,4X,39'0 DO YOU HAVE A PREDETERMINED SET OF APPROXIMATING
1FUNCTION CONSTRAINTS?,/)
      READ*,MYOPT2
      IF(MYOPT2 .EQ. 1) GO TO 70
50 PRINT 1
60 FORMAT (/,4X,39'0 ENTER THE 1-D FREQUENCY THAT YOU WISH TO MAP TO Y
1OUR 2-D CONTOUR /,4X,21H 1. TO 1.0 (X PI)),/
      READ*,F
      IF(F .LE. 3 .OR. F .GE. 1) GO TO 5
      COSW = COS(F*PI)
      PRINT 8
80 FORMAT (/,4X,39'0 DO YOU HAVE A PREDETERMINED SET OF CONSTRAINT EQU
1ATIONS?,/)
      READ*,MYOPT4
      IF(MYOPT4 .EQ. 1) GO TO 110
90 PRINT 1
100 FORMAT (/,4X,39'0 CHOOSE ONE OF THE FOLLOWING CONSTRAINTS BY NUMBER
1.,/,6X,28H1) NE1 MAPS TO (W2,W1)=(0,0),6X,
2      31H2) W=PI MAPS TO (W2,W1)=(PI,PI),/,6X,
3      21H3) BOTH 1 AND 2 ARE ONE /)
      READ*,MYOPT5
      IF(MYOPT5 .EQ. 1 .OR. MYOPT5 .EQ. 2) NCON = 1
      IF(MYOPT5 .EQ. 3) NCON = 2
C
C IF THE PROGRAM SUPPORTS THE CONSTRAINTS, THE LOWER LEFT PART OF THE A ARRAY
C AND THE LOWER PART OF THE B ARRAY ARE SET UP ASSUMING THAT THE CONSTRAINT W=1
C MAPS TO (W2,W1)=(PI,PI) WAS CHOSEN BY THE USER.
C
      B(MYOPT2+1) = 1.
      DO 107 K=1,NCON
          DO 104 J=1,MYOPT2
              A(MYOPT2+K,J) = 1.0
104      CONTINUE
107      CONTINUE
      IF(MYOPT5 .EQ. 1) GO TO 155
C
C IF THE USER CHOOSES OTHER PROGRAM GENERATED CONSTRAINTS, ONLY A FEW ENTRIES
C OF THE A AND B ARRAYS MUST BE CHANGED.
C
      B(MYOPT2+NCON) = -1.0
      A(MYOPT2+NCON,2) = -1.0
      A(MYOPT2+NCON,3) = -1.0
      IF(MYOPT2 .NE. 9) GO TO 155
      A(MYOPT2+NCON,5) = -1.0
      A(MYOPT2+NCON,6) = -1.0
      GO TO 155
C
C HERE THE USER CAN ENTER HIS OWN CONSTRAINT EQUATIONS. EACH CONSTRAINT
C EQUATION USES ONE ROW OF THE A ARRAY (LOWER LEFT SECTION) AND ONE ENTRY OF
C THE B ARRAY (LOWER SECTION).
C
110 PRINT 120
120 FORMAT (/,4X,41'ENTER THE NUMBER OF CONSTRAINT EQUATIONS.,/)
      READ*,NCON
      IF(NCON .LT. 1 .OR. NCON .GT. MYOPT2) GO TO 110
      IF(MYOPT2 .EQ. 4) PRINT 130
130 FORMAT (/,4X,54' EACH CONSTRAINT EQUATION MUST HAVE THE FOLLOWING
1 FORM: /,4X,55H COS(W) = A + B*COS(W1) + C*COS(W2) + D*COS(W1)*COS
2 (W2). /,4X,64H ENTER COS(W), 1, COS(W1), COS(W2), COS(W1)*COS(W2),
3 AND CARPIGE /,4X,76H RETURN FOR EACH CONSTRAINT EQUATION.,/)

```

IF(MYOPT2 .EQ. 51) PRINT 140

140 FORMAT (/ ,4X,53H>0S(W) = A + B*COS(W1) + C*COS(W2) + D*COS(W1)*COS(W2) +
E*COS(W1)*COS(W2) + F*COS(W1)*COS(W2) + G*COS(W1) + H*COS(W2) + ,/ ,4X,56H>0S(W1)*COS(W2) +
4ENTER COS(W), 1, COS(W1), COS(W2), / ,4X, 52H>0S(W1)*COS(W2), COS(W
51)*COS(W2), COS(W1)*COS(W2), COS(W1), / ,4X,57H>0S(W2), COS(W1)*COS(W2),
5)*COS(W2), AND CARRIAGE RETURN FOR EACH EQUATION.,/)

150 DO 170 LREAD=1,NCON
150 READ*,B(MYOPT2+LREAD), (A(MYOPT2+LREAD,JZ), JZ=1,NFOPT2)

THE UPPER RIGHT ENTRIES OF THE A ARRAY ARE ENTERED. EACH CONSTRAINT EQUATION
USES ONE COLUMN IN THIS SECTION OF THE A ARRAY.

155 DO 170 KP=1,NCON
155 DO 160 JP=1,MYOPT2
155 B(JP,MYOPT2+KP) = A(MYOPT2+KP,JP)
160 CONTINUE
170 CONTINUE

THE LOWER RIGHT SECTION OF THE A ARRAY IS ZERO FILLED.

190 DO 210 NA=1,NCON
190 DO 191 NB=1,NCON
190 A(MYOPT2+NA,MYOPT2+NB) = 0.0
191 CONTINUE
200 CONTINUE

HERE THE COEFFICIENTS OF THE CONSTRAINT EQUATIONS ARE SAVED SO THAT
THEY CAN BE PRINTED LATER.

204 DO 215 LSA=1,NCON
204 D(-SA,1) = B(MYOPT2 + LSA)
204 DO 214 LSB=1,MYOPT2
204 D(LSA ,LSB+1) = A(MYOPT2+LSA,LSB)
205 CONTINUE
205 CONTINUE

HERE THE USER CAN ENTER A SET OF POINTS (UP TO 500) TO DEFINE HIS CONTOUR
IN THE W2,W1 PLANE.

210 PRINT 210
210 FORMAT (/ ,4X,67HDO YOU WISH TO ENTER A SET OF POINTS THAT DEFINES
1 YOUR CONTOUR?,/)
210 READ*,MYOPT6
210 IF(MYOPT6 .NE. 1) GO TO 250
210 PRINT 220

220 FORMAT (/ ,4X,37!HOW MANY POINTS DO YOU WISH TO ENTER?,/)
220 READ*,NPOINTS
220 PRINT 230

230 FORMAT (/ ,4X,53H>ENTER THE (W2,W1) COORDINATE PAIRS (W2 IS THE HORIZONTAL
230 AXIS).,/,4X,32H>CARRIAGE RETURN WHEN CONVENIENT.,/)

230 DO 240 NP=1,NPOINTS
230 READ*, W2(NP), W1(NP)
240 CONTINUE
240 GO TO 270

HERE THE USER CAN ENTER HIS SET OF POINTS BY USING A SUBROUTINE THAT CONTAINS
THE EQUATION THAT DEFINES HIS CONTOUR IN THE W2,W1 PLANE.

250 PRINT 250
260 FORMAT (/ ,4X,54H>IF YOU HAVE A CASE NUMBER ENTER IT: OTHERWISE ENT
260 1ER 0.,/)
260 READ*,NCASE
260 CALL YOURSUB(NCASE,NPOINTS,W2,W1)

```

C THE UPPER LEFT SECTION OF THE A ARRAY IS FILLED USING EQUATIONS OF THE FORM
C
C A(KAB,KAC) = A(KAC,KAB) = SUM: G (W2 ,W1 )*G (W2 ,W1 )
C                                     IM=1   KAB   IM   IM   KAC   IM   IM
C
C 270      DO 275 LN=1,NPOINTS
C           G(1,LN) = 1.0
C           G(2,LN) = COS(W1(LN)*PI)
C           G(3,LN) = COS(W2(LN)*PI)
C           G(4,LN) = G(2,LN) * G(3,LN)
C           IF(MYOPT2 .EQ. 4) GO TO 275
C           G(5,LN) = COS(W1(LN) *2 * PI)
C           G(6,LN) = COS(W2(LN) *2 * PI)
C           G(7,LN) = G(2,LN) + G(8,LN)
C           G(8,LN) = G(7,LN) * G(3,LN)
C           G(9,LN) = G(7,LN) + G(8,LN)
C
C 275      CONTINUE
C
C 280      DO 300 KAB=1,MYOPT2
C           DO 290 KAC=KAB,MYOPT2
C           A(KAB,KAC) = 1.0
C           DO 280 IM=1,NPOINTS
C           A(KAB,KAC) = A(KAB,KAC) + G(KAB,IM) * G(KAC,IM)
C
C 290      CONTINUE
C
C 300      CONTINUE
C
C THE UPPER SECTION OF THE B ARRAY IS FILLED USING EQUATIONS OF THE FORM:
C
C           NPOINTS
C           B(LAM) = SUM: COS(W)*G (W2 ,W1 )
C           LAB=1           LAM   IM   IM
C
C           DO 320 LAM=1,MYOPT2
C           B(LAM) = 0.
C           DO 310 LAB=1,NPOINTS
C           B(LAM) = B(LAM) + COSW * G(LAM,LAB)
C
C 310      CONTINUE
C
C 320      CONTINUE
C
C THE MATRIX EQUATION A*CURVFit = CURVFit IS SOLVED FOR THE CURVFit ARRAY.
C
C
C           NEQ = MYOPT2 + NCON
C           CALL LINEQ(A,B,CURVFit,NEQ)
C           IF(NEQ .GT. 0) GO TO 360
C           PRINT 325
C
C 325      FORMAT (/,4X,67HTHE MATRIX EQUATION DOES NOT HAVE A UNIQUE SOLUTION
C           10N WITH THE SET OF,/,4X,69HPOINTS CHOSEN. ENTER 1 TO START OVER OR
C           2R 2 TO TERMINATE THIS PROGRAM.,/)
C           READ*,MYOPT9
C           IF(MYOPT9 .EQ. 1) GO TO 17
C           GO TO 450
C
C 330      IF(MYOPT2 .EQ. 4) PRINT 340
C
C 340      FORMAT (/,4X,47HTHE APPROXIMATING FUNCTION HAS THE FORM: / ,4X,
C           1      57HP(W2,W1) = A + B*COS(W1) + C*COS(W2) + D*COS(W1)*COS(W2).
C           2      ,/,4X,19HENTER A THROUGH D.,/)
C           IF(MYOPT2 .EQ. 9) PRINT 350
C
C 350      FORMAT (/,4X,47HTHE APPROXIMATING FUNCTION HAS THE FORM: / ,4X,
C           1      58HP(W2,W1) = A + B*COS(W1) + C*COS(W2) + D*COS(W1)*COS(W2) +
C           2      ,/,4X,57HE*COS(W1)*COS(2W2) + F*COS(2W1)*COS(W2) + G*COS(2W1) +
C           3      H*COS(2W2) + ,/,4X,40HI*COS(2W1)*COS(2W2). ENTER A THROUGH I.,/)
C           READ*, (CURVFit(TCUR),ICLR=1,MYOPT2)
C
C ONCE THE APPROXIMATING CONSTANTS ARE FOUND, THE MODEL-AN TRANSFORMATION
C IS EVALUATED ON A 441 POINT GRID IN THE W2,W1 PLANE TO SEE IF THE MAPPING
C FROM THE W-AXIS TO THE W2,W1 PLANE WILL BE WELL-DEFINED.

```

```

C
360      DO 380 LZ=1.21
          LC=22-L7
          WA = PI * (LC-1)/20.
          DO 370 LW=1,21
          LD = 22-LW
          WB = PI * (LD-1)/20.
          1  IF(MYOPT2 .EQ. 4) CHECK = CURVFIT(1) +
          2  CURVFIT(2)*COS(WA) + CURVFIT(3)*COS(WB) +
          2  CURVFIT(4)*COS(WA)*COS(WB)
          1  IF(MYOPT2 .EQ. 9) CHECK = CURVFIT(1) +
          2  CURVFIT(2)*COS(WA) + CURVFIT(3)*COS(WB) +
          2  CURVFIT(4)*COS(WA)*COS(WB) + CURVFIT(5)*COS(WA)*
          3  COS(2*WB) + CURVFIT(6)*COS(2*WA)*COS(WB) + CURVFIT(7)
          4  *COS(2*WA) + CURVFIT(8)*COS(2*WB) + CURVFIT(9)
          5  *COS(2*WA)*COS(2*WB)
          IF(ABS(CHECK) .GT. 1) GO TO 390
370      CONTINUE
380      CONTINUE

```

```

1  GO TO 455
390 PRINT 400
400 FORMAT (/,4X,64HYOUR CONTOUR WITH YOUR FREQUENCY PRODUCES AN ILL-
1  DEFINED MAPPING,/,4X,63H FROM THE W-AXIS TO THE W2,W1 PLANE. ENTER
2  ONE OF THE FOLLOWING,/,4X,15HOPTION NUMBERS.,/,6X,
3  49H1) CHOOSE DIFFERENT PROGRAM GENERATED CONSTRAINTS,/,5X,
4  38H2) ENTER YOUR OWN CONSTRAINT EQUATIONS,/,6X,
5  51H3) ENTER A SET OF APPROXIMATING FUNCTION CONSTANTS,/,5X,
5  13H4) START OVER,5X,14H5) TRY SCALING,5X,
7  25H5) TERMINATE THIS PROGRAM,/)
READ*,MYOPT7
IF(MYOPT7 .EQ. 1) GO TO 91
IF(MYOPT7 .EQ. 2) GO TO 110
IF(MYOPT7 .EQ. 3) GO TO 331
IF(MYOPT7 .EQ. 4) GO TO 16
IF(MYOPT7 .EQ. 5) GO TO 450

```

```

C
C IF THE MAPPING FROM THE W-AXIS TO THE W2,W1 PLANE IS ILL-DEFINED, THE USER M
C WISH TO USE THE SCALING ROUTINE. IT WILL USUALLY PRODUCE A WELL-DEFINED
C MAPPING. HOWEVER, THE SCALED FREQUENCY MAY NOT EQUAL (OR EVEN BE CLOSE TO)
C THE VALUE OF THE USERS ORIGINAL FREQUENCY.
C

```

```

FMAX = -1.0E+19
FMIN = 1.0E+19
NTRY = 101
NK = 100
DO 430 I=1,1TRY
    DO 420 J=1,NTRY
        IF(MYOPT2 .NE. 4) GO TO 410
        S = CURVFIT(1) + CURVFIT(2)*COS(PI*(I-1)/NK) +
        1  CURVFIT(3)*COS(PI*(J-1)/NK) + CURVFIT(4)*
        2  COS(PI*(I-1)/NK)*COS(PI*(J-1)/NK)
        GO TO 418
410      S = CURVFIT(1) + CURVFIT(2)*COS(PI*(I-1)/NK) +
        1  CURVFIT(3)*COS(PI*(J-1)/NK) + CURVFIT(4)*
        2  COS(PI*(I-1)/NK)*COS(PI*(J-1)/NK) + CURVFIT(5)
        3  *COS(PI*(I-1)/NK)*COS(2*PI*(J-1)/NK) +
        4  CURVFIT(6)*COS(2*PI*(I-1)/NK)*COS(PI*(J-1)/NK) +
        5  CURVFIT(7)*COS(2*PI*(I-1)/NK) + CURVFIT(8)*
        6  COS(2*PI*(J-1)/NK) + CURVFIT(9)*COS(2*PI*(I-1)/NK)*
        7  COS(2*PI*(I-1)/NK)
418      FMAX = AMAX1(FMAX,S)
        FMIN = AMIN1(FMIN,S)
420      CONTINUE
430      CONTINUE

```

```

C1 = 2.0/(FMAX-FMIN)
C2 = C1*FMAX - 1.0

```

```

CURVFIT(1) = C1*CURVFIT(1) - C2
CURVFIT(2) = C1*CURVFIT(2)
CURVFIT(3) = C1*CURVFIT(3)
CURVFIT(4) = C1*CURVFIT(4)
IF(MYOPT2 .EQ. 1) GO TO 441
CURVFIT(5) = C1*CURVFIT(5)
CURVFIT(6) = C1*CURVFIT(6)
CURVFIT(7) = C1*CURVFIT(7)
CURVFIT(8) = C1*CURVFIT(8)
CURVFIT(9) = C1*CURVFIT(9)
440 WC = ACOS(C1*COSW - C2)/PI
F = WS
PRINT 451,WS
451 FORMAT (/,1X,42HY CUR ORIGINL FREQUENCY HAS BEEN SCAL ED TO ,
1 F9.6,15HPT. THE SCALED,/,1X,41HFREQUENCY WILL BE MAPPED TO YO
2 UR CONTOUR.,/,1X,42HENT EP ONE OF THE FOLLOWING OPTION NUMBERS.
3 ,/,1X,13H1) START OVER,EX,2FH2) TERMINATE THIS PROGRAM,EX,
4 11H3) CONTINUE,/)
READ*,MYOPT1
IF(MYOPT1 .EQ. 1) GO TO 11
IF(MYOPT1 .EQ. 2) GO TO 452
453 IF(MYOPT2 .EQ. 1) GO TO 457
C HERE THE MOCELLAN TRANSFORMATION IN THE FORM OF MULTIPLE ANGLE COSINE
C FUNCTIONS IS CONVERTED TO THE MOCELLAN TRANSFORMATION IN THE FORM OF
C SINGLE ANGLE COSINE FUNCTIONS RAISED TO THE ZERO, FIRST, AND SECOND POWERS.
C
MOTRAN(1,1) = CURVFIT(1)
MOTRAN(2,1) = CURVFIT(2)
MOTRAN(1,2) = CURVFIT(3)
MOTRAN(2,2) = CURVFIT(4)
CURVFIT(5) = 1.
CURVFIT(6) = 1.
CURVFIT(7) = 1.
CURVFIT(8) = 1.
CURVFIT(9) = 1.
GO TO 458
457 MOTRAN(1,1) = CURVFIT(1) - CURVFIT(7) - CURVFIT(8) + CURVFIT(9)
MOTRAN(2,1) = CURVFIT(2) - CURVFIT(5)
MOTRAN(1,2) = CURVFIT(3) - CURVFIT(6)
MOTRAN(2,2) = CURVFIT(4)
MOTRAN(2,3) = CURVFIT(5) * 2
MOTRAN(3,2) = CURVFIT(5) * 2
MOTRAN(3,1) = CURVFIT(7) * 2 - CURVFIT(9) * 2
MOTRAN(1,3) = CURVFIT(8) * 2 - CURVFIT(9) * 2
MOTRAN(3,3) = CURVFIT(9) * 4
C PROGRAM OUTPUT SECTION
C
458 IF(MYOPT2 .EQ. 4) PRINT 470, (CURVFIT(J6),J6=1,MYOPT2)
IF(MYOPT2 .EQ. 1) WRITE(1,470) (CURVFIT(J6),J6=1,MYOPT2)
IF(MYOPT2 .EQ. 0) PRINT 480, (CURVFIT(J8),J8=1,MYOPT2)
IF(MYOPT2 .EQ. 9) WRITE(1,480) (CURVFIT(J9),J9=1,MYOPT2)
IF(MYOPT2 .EQ. 11) GO TO 460
DO 453 L11=1,NOON
IF(MYOPT2 .EQ. 4) PRINT 490,(0(L11,M11),M11=1,5)
IF(MYOPT2 .EQ. 4) WRITE(1,490) (0(L11,M11),M11=1,5)
IF(MYOPT2 .EQ. 9) PRINT 500,(0(L11,M11),M11=1,10)
IF(MYOPT2 .EQ. 9) WRITE(1,500) (0(L11,M11),M11=1,10)
459 CONTINUE
PRINT 510,F
WRITE (1,510) =
460 CONTINUE
470 FORMAT (// ,1X,1(14+),/ ,11X,51HLINEAR LEAST SQUARES APPROXIMATI
ON WITH CONSTRAINTS,/,1X,71HTHE APPROXIMATION HAS THE FORM: F(W
22,W1) = A + B*COS(W1) + C*COS(W2) + ,/ ,1X,22HD*COS(W1)*COS(W2) WI

```

SUBROUTINE LINE0(A,B,X,N)

THIS SUBROUTINE SOLVES UP TO 18 SIMULTANEOUS LINEAR ALGEBRAIC EQUATIONS BY GAUSS-JORDAN ELIMINATION

THE CALLING PROGRAM MUST SPECIFY THE FOLLOWING:
THE COEFFICIENT MATRIX (A)
THE RIGHT-HAND VECTOR (B)
THE NUMBER OF EQUATIONS (N)

THE SUBROUTINE WILL RETURN THE SOLUTION VECTOR (X) TO THE CALLING PROGRAM

```

DIMENSION A(13,17),B(18),X(18)
C THE FOLLOWING DO LOOP REFERS TO EACH OF THE FIRST N COLUMNS
DO 4 I=1,N
C THE FOLLOWING DO LOOP REFERS TO EACH ROW PER COLUMN EXCEPT THE PIVOT
C ROW
DO 2 K=1,N
  IF(K .EQ. I) GO 9 2
  IF(A(I,I) .EQ. 0.) GO TO 6
  CONST = -A(K,I)/A(I,I)
C THE FOLLOWING DO LOOP REFERS TO EACH ELEMENT IN A ROW
DO 3 J=1,N
  A(K,J) = A(K,J) + CONST * A(I,J)
  IF(J .EQ. I) A(K,J) = 0.
1  CONTINUE
  B(K) = B(K) + CONST * B(I)
2  CONTINUE
  CONST = A(I,I)
C THE FOLLOWING DO LOOP REFERS TO EACH ELEMENT IN THE PIVOT ROW
DO 3 J=1,N
3  A(I,J) = A(I,J)/CONST
  A(I,I) = 1.
  B(I) = B(I)/CONST
4  CONTINUE
C OBTAIN SOLUTION VECTOR
DO 5 I=1,N
5  X(I) = B(I)
6  IF(A(I,I) .EQ. 0.) N=-3
  RETURN
END

```

OVERLAY(MCL, 2, 1)

C PROGRAM GRAPH

C THIS PROGRAM PROVIDES THE DATA NECESSARY FOR THE CALCOMP PLOTTER TO PLOT
C THE ONE-DIMENSIONAL MAGNITUDE VERSUS FREQUENCY (W) CURVE, THE TWO-
C DIMENSIONAL CONTOUR CURVES, AND THE TWO-DIMENSIONAL MAGNITUDE VERSUS
C FREQUENCY (W2,W1) THREE-DIMENSIONAL FIGURE.

C COMMON /AAA/HDCHE3 /RER/CURVFIT /CCC/MYOPT2 /ONE/NFILT

C DIMENSION CURVFIT(14), HCOOR(1,2), VCOOR(102), HDCHE3(57)

C THE FIRST NINE ELEMENTS OF CURVFIT STORE THE NINE CONSTANTS OF THE
C APPROXIMATING FUNCTION. THE FIRST 14 LOCATIONS OF HCOOR AND VCOOR STORE
C THE HORIZONTAL AXIS AND VERTICAL AXIS COORDINATES OF THE POINTS TO BE
C PLOTTED. HDCHE3 IS DIMENSIONED ((NEMAX-1)/2+1).

C PI = 3.1415926535898

C THE USER CAN CHOOSE TO HAVE TABULAR DATA (UP TO 100 POINTS) FOR ANY CONTOUR IN
C THE FIRST QUADRANT OF THE W2,W1 PLANE PRINTED ON HIS HARDCOPY OUTPUT FILE
C (RESULT). ALL CONTOURS ARE FOUR QUADRANT SYMMETRIC. THE PROGRAM MAY GENERATE
C INVALID POINTS. ALL INVALID POINTS HAVE THE COORDINATES (-.15,-.25) AND
C SHOULD BE DISREGARDED BY THE USER.

C PRINT 1

10 FORMAT (/,4X,67'DO YOU WISH TO HAVE TABULAR DATA FOR ANY CONTOURS
1 IN THE FIRST ,/,1X,58'HQUADRANT OF THE W2,W1 PLANE PRINTED OUT O
2N YOUR HARDCOPY OUTPUT FILE ,/,4X,9H(RESULT)?,/)

READ*,MYOPT2

IF(MYOPT2 .NE. 1) GO TO 50

C CURVFIT(1) = 1. TELLS THE REST OF THE PROGRAM THAT TABULAR DATA IS
C TO BE GENERATED AND NOT PLOTS.

C CURVFIT(1) = 1.

15 PRINT 2

20 FORMAT (/,4X,51'ENTER N (1.0 TO 1.0) OR ENTER 2 TO CONTINUE WITH
1THE PROGRAM.,/)

READ*,FOPT

IF(FOPT .EQ. 2) GO TO 50

IF(FOPT .LT. 1.0 .OR. FOPT .GT. 1) GO TO 15

COSWOPT = COS(PI * FOPT)

PRINT 3

30 FORMAT (/,4X,7'HOW MANY POINTS DO YOU WANT OUTPUT? ,/)

READ*,MYOPT22

IF(MYOPT22 .GT. 100) MYOPT22 = 100

C SUBROUTINE FORDER OR SCORDER GENERATE THE MYOPT22 POINTS TO BE
C OUTPUT TO THE FILE RESULT.

C IF (MYOPT2 .EQ. 1) CALL FORDER(MYOPT22, HCOOR, VCOOR, CURVFIT,
1 COSWOPT)
IF(MYOPT2 .EQ. 2) CALL SCORDER(MYOPT22, HCOOR, VCOOR, CURVFIT,
1 COSWOPT)

C HERE THE POINTS ARE OUTPUT TO THE FILE RESULT.

C WRITE(1,41) FOPT

PRINT 41,FOPT

41 FORMAT (//,1X,8:FOR W = ,F7.4,14HPI, (W2,W1) =:)

LS2 = 1

LE2 = 14

46 IF(MYOPT22 .LE. LE2) LE2 = MYOPT22

WRITE(1,51) (HCOOR(N),N=LS2,LE2)

```

PRINT 11,(HCOOR(1,1),VCOOR(1,1),N1=LS2,LE2)
WRITE(1,55) (VCOOR(1,1),N1=LS2,LE2)
PRINT 11,(VCOOR(1,1),N1=LS2,LE2)
IF(MYOPT22.NE.152) WRTTF(1, 7)
IF(MYOPT22.NE.152) PRTNT 57
51 FORMAT (4X, SHW2: ,14(F7.4,2X))
56 FORMAT (4X, SHW1: ,14(F7.4,2X))
57 FORMAT (2X, /)
LS2 = LS2 + 14
LE2 = LE2 + 14
IF(LE2.LE. MYOPT22) GO TO 56
GO TO 15
55 WRITE (1,19)
PRINT 58
58 FORMAT (/,X,"(N1 POINTS WITH COORDINATES (-.15,-.25) ARE INVALID"
1 " AND SHOULD BE DISREGARDED)",/)
C
C THE USER CAN CHOOSE TO GENERATE THE POINTS NECESSARY FOR THE ONE-DIMENSIONAL
C MAGNITUDE VERSUS FREQUENCY (W) CURVE AND THE TWO-DIMENSIONAL CONTOUR CURVES.
C
66 PRINT 7.
71 FORMAT (/,4X,5) HOW YOU WANT TO CREATE A PLOT FILE FOR THE CALCOMP
1P PLOTTER?,/
READ*,MYOPT23
IF(MYOPT23.NE.1) GO TO 11.
C
C CURVFIT(1) = 2.0 TELLS THE REST OF THE PROGRAM THAT PLOTS ARE TO
C BE GENERATED AND NOT TABULAR DATA.
C
C CURVFIT(1) = 2.0
C
C THE PROGRAM PLOTS NPTS EQUALLY SPACED POINTS FOR THE ONE-DIMENSIONAL MAGNITUDE
C VERSUS FREQUENCY (W) CURVE.
C
NPTC = 15
NINTVL = 99
KD = ((NFTLT-1)/2) + 1
C
C THE ONE-DIMENSIONAL MAGNITUDE FOR EACH W IS CALCULATED USING THE EQUATION
C
C MAG(H(W)) = SUM: H100HER(J) (COSH) .
C J=0
C
DO 80 MA=1,NPTS
W = PI * (MA-1)/NINTVL
HCOOR(MA) = W/PI
VCOOR(MA) = H100HER(1)
DO 80 J=2,KD
PROD = H100HER(MB) * (COS(W) ** (MB-1))
VCOOR(MA) = VCOOR(MA) + PROD
CONTINUE
80
90 CONTINUE
C
C THESE SUBROUTINE CALLS TO THE CALCOMP PLOTTER GENERATE THE PLOT.
C
HCOOR(1,1) = 1.0
HCOOR(1,2) = .15
VCOOR(1,1) = -.25
VCOOR(1,2) = .25
CALL PLOT(0.0,-1.0,-3)
CALL PLOT(0.0,1.25,-3)
CALL AXIS(0.,1.,'3HW-AXIS (Y PI),-13,7.,0.,HCOOR(1,1),HCOOR(1,2))
CALL AXIS(0.,0.0,28HFREQUENCY RESPONSE MAGNITUDE,28,7.,30.,
1 VCOOR(1,1),VCOOR(1,2))
CALL LINE(HCOOR,VCOOR,1^0,1,0,1)
CALL SYMBOL (.188,7.5,.21,32H ONE-DIMENSIONAL PROTOTYPE FILTER,0.,)

```

```

1 32)
CALL FLOT(2,0,L,1E,-3)

C THE PROGRAM PLOTS 11 TWO-DIMENSIONAL CONTOURS. FOR THESE CONTOURS, W RANGES
C FROM .01 TO 1.0 WITH A .1 INTERVAL.
C
NCNT = 11
NCTIVL = 10

C THESE SUBROUTINE CALL'S TO THE CALCOMP PLOTTER GENERATE THE PLOT (SUBROUTINE
C FORDR OR SDORDER GENERATE THE MPOINT27 POINTS TO BE PLOTTED).
C
MPOINT27 = 100
CALL FLOT(1.0,-5.0,-3)
CALL FLOT(0.0,2.0,-3)
CALL AXIS(0.0,1.0,14H42-AXIS (X PI),-14,7.0,3.0,.01,.15)
CALL AXIS(0.0,1.0,14H41-AXIS (X PI),14,7.0,3.05,.01,.15)
CALL SYMBOL(1.03,7.05,.21,24H1WC-DIMENSIONAL CONTOURS,.03,24)
DO 1 3 MN=1,NCNT
COSWOPT = COS(PI * (MN-1)/NCTIVL)
IF (MPOINT27.EQ. 0) CALL FORDR(MPOINT27, HCOOR, VCOOR, CURVFIT
1 ,COSWOPT)
IF (MPOINT27.EQ. 9) CALL SDORDER(MPOINT27, HCOOR, VCOOR, CURVFIT
1 ,COSWOPT)
100 CONTINUE
CALL FLOT(10.0, 0.0,-3)
CALL FLOT(0.0)
111 CONTINUE

C IN THIS SECTION THE DATA NECESSARY FOR A DISSPLA 3-D PLOT OF THE
C MAGNITUDE OF THE 2-D FREQUENCY RESPONSE IS PUT ON TAPE2.
C
KD = ((NFILT - 1)/2 + 1)
PRINT 271
271 FORMAT(//,4X,57H0 YOU WANT TO CREATE A DISSPLA 3-D PLOT OF THE FR
EQUENCY RESPONSE,/)
READ*,MPOINT1
IF(MPOINT1.NE.1) 10 TO 310
WRITE(2,275) K7
275 FORMAT(I3)
WRITE(2,276) (CURVFIT(MT),MT=1,9)
280 FORMAT(E18.3,/,4E18.3)
WRITE(2,280)(H1*4E9(MT),MT=1,KD)
290 FORMAT(9(7E18.3,/,))
PRINT 310
303 FORMAT(/,8X,55H THE DATA REQUIRED FOR THE DISSPLA PLOT IS NOW ON T
APE2,//)
310 CONTINUE
END

C -----
C SUBROUTINE FORDR(MPOINT22,HCOOR,VCOOR,CURVFIT,COSWOPT)
C
C THIS SUBROUTINE CALCULATES THE POINTS THAT ARE OUTPUT TO FILE RESULT OR
C PLOTTED AS TWO-DIMENSIONAL CONTOURS IF THE FIRST ORDER MOCELLAN
C TRANSFORMATION IS USED.
C
C MPOINT22 - THE NUMBER OF POINTS TO BE OUTPUT OR PLOTTED
C HCOOR AND VCOOR - STORE THE HORIZONTAL AXIS AND VERTICAL AXIS
C COORDINATES OF THE POINTS TO BE OUTPUT OR PLOTTED
C CURVFIT - THE FIRST FOUR LOCATIONS STORE THE FOUR TERMS OF THE
C APPROXIMATING FUNCTION
C COSWOPT - THE ONE-DIMENSIONAL FREQUENCY (W) TO BE MAPPED TO A
C TWO-DIMENSIONAL CONTOUR
C
C DIMENSION HCOOR(102), VCOOR(102), CURVFIT(18), INK(100)
C

```

REAL CNTF-VR

C PI = 3.141592653 198
DO 5 MP=2,100
5 INK(N2) = 3
FORT = 4COS(COSW PT1)/2T
MPOFT1 = .01
MPOFT2 = .001
C FOR ANY W, THE LARGEST AND SMALLEST VALID W2 IS FOUND BY EVALUATING THE
C FIRST ORDER MODEL LAW TRANSFORMATION EQUATION, SOLVED FOR W1, AT MANY
C EQUALLY SPACED VALUES OF W2
C
C W2ST = PI
C W2END = 0.
C DO 1 I=1, MPOFT1
C W2 = PI * (-1.0)/MPOFT2
C IF((CUPVFIT(2)+CURVFIT(4)*COS(W2)) .EQ. 0.) GO TO 1
C COSW1 = (COS WOPT-CUPVFIT(1)-CURVFIT(3)*COS(W2))/
1 (CURVFIT(2)+CURVFIT(4)*COS(W2))
C IF(ABS(COSW1) .GT. 1) GO TO 1
C W2ST = AMIN1(W2ST, W2)
C W2END = AMAX1(W2END, W2)
10 CONTINUE
C
C THE INTERVAL USED TO EQUALLY SPACE THE W2 VALUES IS CALCULATED.
C
C INTERVAL = (W2END-W2ST)/(MPOFT2-1)
C
C THE MPOFT2 (W2,W1) COORDINATE PAIRS ARE
C CALCULATED ASSUMING ALL VALUES OF W2 BETWEEN THE SMALLEST AND LARGEST VALID
C W2 ARE ALSO VALID.
C
C W2 = W2ST
C DO 4 J=1, MPOFT2
C HCOP(J) = W2/PI
C IF((CUPVFIT(2)+CURVFIT(4)*COS(W2)) .EQ. 0.) GO TO 35
C COSW1 = (COS WOPT-CUPVFIT(1)-CURVFIT(3)*COS(W2))/
1 (CURVFIT(2)+CURVFIT(4)*COS(W2))
C IF(ABS(COSW1) .GT. 1) GO TO 35
C W1 = 4COS(COSW1)
C VCOORD(J) = W1/PI
C GO TO 38
C
C IF A VALUE OF W2 BETWEEN THE SMALLEST AND LARGEST VALID W2 IS NOT VALID, THE
C COORDINATES OF THE POINT CONTAINING THE INVALID W2 ARE ARBITRARILY SET TO
C (-.15, -.25). THESE POINTS SHOULD BE DISREGARDED BY THE USER.
C
C 35 HCOP(J) = -.15
C VCOORD(J) = -.25
C INK(J) = 3
C INK(J+1) = 1
C 38 W2 = W2 + INTERVAL
40 CONTINUE
C IF(CUPVFIT(10) .EQ. 1.0) RETURN
C
C THESE CALLS TO THE CILCOOP PLOTTER CAUSE THE CURVE TO BE PLOTTED.
C
C X = HCOP(1)/.15
C Y = VCOORD(1)/.15
C CALL PLOT(X,Y,3)
C IF(X .LT. 0.) INK(2) = 3
C IF(X .GE. 0.) CALL PLOT(X,Y,2)
C DO 5 MP=2,100
C X = HCOP(MP)/.15
C Y = VCOORD(MP)/.15

```

      CALL PLOT(Y, "7-NR(MF))
      CONTINUE
      IF(X .GE. 0.) CALL PLOT(Y, Y, 2)
      Y1 = HCO0(1)/.1 + .12
      Y1 = VCO0(1)/.1 + .12
      IF(Y1 .LT. 1.) CALL NM355(X1, Y1, .1), FOPT, J, 1
      X2 = HCO0(1)/.15 + .12
      Y2 = VCO0(1)/.15 + .12
      IF(X2 .LT. 1.) CALL NM355(Y2, Y2, .1), FOPT, J, 1
      RETURN
      END

C -----
C -----
C      SUBROUTINE SP222(MYOPT22, HCO0, VCO0, CURVFIT, COSWOPT)
C
C THIS SUBROUTINE CALCULATES THE POINTS THAT ARE OUTPUT TO FILE RESULT 02
C PLOTTED AS FW,-DIRECTIONAL CONTOURS OF THE SECOND ORDER MCCLELLAN
C TRANSFORMATION. IS USED.
C
C MYOPT22 = THE NUMBER OF POINTS TO BE OUTPUT OR PLOTTED
C HCO0 AND VCO0 = STORE THE HORIZONTAL AXIS AND VERTICAL AXIS COORDINATES
C OF THE POINTS TO BE OUTPUT OR PLOTTED
C CURVFIT = THE EIGHT WIRE LOCATIONS STORE THE NINE TERMS OF THE
C APPROXIMATING FUNCTION
C COSWOPT = THE 04 - DIMENSIONAL FREQUENCY TO BE MAPPED TO A
C TWO-DIMENSIONAL CONTOUR
C
C      DIMENSION HCO0(12), VCO0(12), CURVFIT(18), INK(1,1)
C
C      REAL INTVL1, INTVL2
C
C      PI = 3.141592653589
C      DO F, NF=2, 12
C      5      INK(F) = 2
C      FUF = ACOS(COSW OPT) * PI
C      MPOPT1 = F11
C      MPOPT2 = F12
C
C      FOR ANY W, THE SECOND ORDER MCCLELLAN TRANSFORMATION EQUATION (A QUADRATIC)
C      IS SOLVED FOR W1 USING THE QUADRATIC FORMULA. FOR THE FIRST SOLUTION OF THE
C      QUADRATIC, THE LARGEST AND SMALLEST VALID W2 ARE FOUND BY EVALUATING THE
C      FIRST SOLUTION AT MANY EQUALLY SPACED VALUES OF W2.
C
C      W2ST1= PI
C      W2END1= 1.4
C      W2ST2= PI
C      W2END2= 1.4
C      DO 2, J=1, MPOPT1
C      2      W2 = PI * (J-1.1)/MPOPT2
C      Z = 2*CURVFIT(1)*COS(W2)+2*CURVFIT(7)+2*CURVFIT(3)*COS(2*W2)
C      " = CURVFIT(2)+CURVFIT(4)*COS(W2)+CURVFIT(5)*COS(2*W2)
C      F = (CURVFIT(1)-CURVFIT(7)-COSWOPT) + (CURVFIT(3)-CURVFIT(6))
C      1      *COS(42) + (CURVFIT(6)-CURVFIT(9))*COS(2*W2)
C      RADICAL = Z**2 - 4*F**2
C      IF(RADICAL .LT. 0.) GO TO 2
C      IF(A .EQ. 0.) COSW1 = -C/P
C      IF(A .NE. 0.) COSW1 = (-B + SQRT(RADICAL)) / (2*A)
C      IF(ABS(COSW1) .GT. 1) GO TO 1
C      W2ST1= AMIN(W2ST1, W2)
C      W2END1= AMAX(W2END1, W2)
C
C      NOW FOR ANY W AND F0 THE SECOND SOLUTION OF THE QUADRATIC EQUATION,
C      THE LARGEST AND SMALLEST VALID W2 IS FOUND BY EVALUATING THE SECOND
C      SOLUTION EQUATION AT MANY EQUALLY SPACED VALUES OF W2.
C
C      10      IF(A .EQ. 0.) COSW1 = -C/P

```

```

IF(I .NE. 1) W2T1 = (-B + SQRT(RADICAL)) / (2*A)
IF(ABS(C0341) .GT. 1) GO TO 2
W2T2 = ATAN(W2T1, B)
W2T3 = ATAN(W2T2, B)

```

2 CONTINUE

C THE VARIABLES USED TO EQUALLY SPACE THE W2 VALUES FOR EACH SOLUTION
C OF THE QUADRANTS FOR THIS ARE CALCULATED.

```

MW0RK1 = MYOPT22/2
MW0RK2 = (MYOPT22/2) + 1
TEST1 = W2T1 - W2T2
TEST2 = W2T2 - W2T3
IF(TEST1 .LT. 1) MW0RK1 =
IF(TEST1 .LT. 1) MW0RK2 = 1
IF(TEST1 .LT. 1) GO TO 61
IF(TEST2 .LT. 1) MW0RK1 = MYOPT22

```

C THE INTERVAL USED TO EQUALLY SPACE THE W2 VALUES IS CALCULATED.

```
1 TVL1 = (W2T1-W2T3) / (MW0RK1-1)
```

C THE MW0RK2/2 (W2,W1) COORDINATE PAIRS ARE CALCULATED ASSUMING THAT
C ALL VALUES OF W2 BETWEEN THE SMALLEST AND LARGEST VALID W2 ARE ALSO VALID.

```

2 W2 = W2T1
DO 5 J1=1,MW0RK1
W00R(J1) = W2/2
A = 2*CURVFIT(1)*COS(W2)+2*CURVFIT(7)+2*CURVFIT(3)*COS(2*W2)
B = CURVFIT(2)+CURVFIT(4)*COS(W2)+CURVFIT(5)*COS(2*W2)
C = (CURVFIT(1)-CURVFIT(4)-COS4OPT) + (CURVFIT(3)-CURVFIT(6))
RADICAL = B*2 + A*A
IF(RADICAL .LT. 1) GO TO 41
IF(A .EQ. 0) COSW1 = -C/B
IF(A .NE. 0) C0341 = (-B + SQRT(RADICAL)) / (2*A)
IF(ABS(C0341) .GT. 1) GO TO 45
W1 = ACOS(C0341)
V00R(J1) = W1/2
GO TO 47

```

C IF A VALUE OF W2 BETWEEN THE VALUES OF THE SMALLEST AND LARGEST VALID W2
C IS NOT VALID, THE COORDINATES OF THE POINT CONTAINING THE INVALID W2
C ARE ARBITRARILY SET TO (-.15,-.25). THESE POINTS SHOULD BE
C DISREGARDED BY THE USER.

```

45 W00R(J1) = -.15
V00R(J1) = -.25
INK(J1) = 3
INK(J1+1) = 7
47 W2 = W2 + TVL1
56 CONTINUE

```

C THESE CALLS TO THE C-L00MP PLOTTER CAUSE THE CURVE TO BE PLOTTED.

```

IF(CURVFIT(18) .EQ. 1) GO TO 60
X = W00R(1)/.15
Y = V00R(1)/.15
CALL PLOT(X,Y,3)
IF(X .LT. 0) TNK(2) = 3
IF(X .GT. 0) TNK(2) = 1
DO 57 M = 1,MW0RK1
X = W00R(M)/.15
Y = V00R(M)/.15
CALL PLOT(X,Y,TNK(M))
57 CONTINUE

```

```

IF(X .GE. 1.) CALL SYMBOL(X,Y,.17,3,1.,-1)
X1 = HOOOR(1)/.1 + .12
Y1 = VOOOR(1)/.1 + .12
IF(X1 .GT. 1.) CALL NUMBER(X1,Y1,.1,1,FOPT,1,1)
X2 = HOOOR(MWORK1)/.15 + .12
Y2 = VOOOR(MWORK1)/.15 + .12
IF(X2 .GT. 1.) CALL NUMBER(X2,Y2,.1,1,FOPT,1,1)
61 IF(TESTC .LT. 1. AND. TEST1 .LT. 1.) GO TO 1

C THE INTERVAL USED TO EQUALLY SELECT THE W2 VALUES IS CALCULATED.
C
C INTVL2 = (W2OPT2-W2OPT1)/(MYOPT22-MWORK1-1)
C
C THE MYOPT22/2 (W2,W1) COORDINATE PAIRS ARE CALCULATED ASSUMING THAT
C ALL VALUES OF W2 BETWEEN THE SMALLEST AND LARGEST VALID W2 ARE ALSO VALID.
C
W2 = W2OPT1
LC 9 J2=MWORK2,MYOPT22
HOOOR(J2) = W2/PT
A = 2*CURVFIT(2)*COS(W2)+2*CURVFIT(7)+2*CURVFIT(3)*COS(2*W2)
B = CURVFIT(2)*CURVFIT(4)*COS(W2)+CURVFIT(5)*COS(2*W2)
C = (CURVFIT(1)-CURVFIT(7)-COS(W2))+(CURVFIT(3)-CURVFIT(1))
D = CURVFIT(8)-CURVFIT(9))*COS(2*W2)
RADICAL = B*B-4*A*C
IF(RADICAL .LT. 1.) GO TO 85
IF(A .EQ. 1.) COSW1 = -C/B
IF(A .NE. 1.) COSW1 = (-B - SQRT(RADICAL))/(2*A)
IF(A*BS(COSW1) .GT. 1.) GO TO 85
W1 = A*COS(COSW1)
VOOC(J2) = W1/PT
GO TO 87

C IF A VALUE OF W2 BETWEEN THE VALUES OF THE SMALLEST AND LARGEST VALID W2
C IS NOT VALID, THE COORDINATES OF THE POINT CONTAINING THE INVALID W2
C ARE ARBITRARILY SET TO (-.17,-.25). THESE POINTS SHOULD BE
C DISREGARDED BY THE USER.
C
85      HOOOR(J2) = -.17
VOOC(J2) = -.25
INR(J2) = 2
INR(J2+1) = 2
87      W2 = W2 + INTVL2
90      CONTINUE
IF(CURVFIT(17) .LT. 1.) GO TO 1.

C THESE CALLS TO THE CIRCLEP PLOTTER CAUSE THE CURVE TO BE PLOTTED.
C
X = HOOOR(MWORK2)/.15
Y = VOOOR(MWORK2)/.15
CALL PLOT(X,Y,2)
IF(Y .LT. 1.) INR(2) = 2
IF(Y .GE. 1.) CALL SYMBOL(X,Y,.17,4,1.,-1)
MW2 = MWORK2 + 1
DO 91 MG=MW2,MWORK22
X = HOOOR(MG)/.15
Y = VOOOR(MG)/.15
CALL PLOT(X,Y,INR(MG))
95      CONTINUE
IF(X .GE. 1.) CALL SYMBOL(X,Y,.17,4,1.,-1)
X1 = HOOOR(MWORK2)/.15 + .125
Y1 = VOOOR(MWORK2)/.15 + .125
IF(X1 .GT. 1.) CALL NUMBER(X1,Y1,.1,1,FOPT,1,1)
X2 = HOOOR(MYOPT22)/.15 + .125
Y2 = VOOOR(MYOPT22)/.15 + .125
IF(X2 .GT. 1.) CALL NUMBER(X2,Y2,.1,1,FOPT,1,1)
101 CONTINUE

```



```

      IF(NBANDS.LE.1 .OR. ITRANS.GT.1.) GO TO 113

C   GRID DENSITY IS 15
C
C   LGR1D=15
C   JRE=NBANDS
C   REDGE(1)=1.0  REDGE(JR)=1.
C   IF(NBANDS.EQ.1) GO TO 123
118   IF(IREPORT.EQ.1.0E.7IREPORT.EQ.3) PRINT 119
119   FORMAT(4X,"ENTER THE BAND EDGE FREQUENCIES FOR "
X ". EACH TRANSIT IN BAND. //")
C   JDM1=JR+1
C   IF(IREPORT.EQ.1.0E.7IREPORT.EQ.3) READ*, (REdge(j), j=2, JR+1)
C   DO 121 J=2, JR+1
C   IF(EDGE(1).GT.1.0E.7*EDGE(j).LT.1.0) GO TO 118
121   CONTINUE
123   CONTINUE
C   DO 124 J=1, JR
C
C   SINCE MCCLELLAN'S PROGRAM USES DEGREES, HERE, RADIANS
C   ARE PARTIALLY CONVERTED TO DEGREES.
C
120   EDGE(J)=REdge(j)/2.
C   IF(IREPORT.EQ.1) PRINT 125
125   FORMAT(4X,"ENTER AN IDEAL ABSOLUTE MAGNITUDE FOR EACH BAND//"
X "(X, 'OF THE PROTOTYPE FILTER (USUALLY 1 OR 0),')//")
C   IF(IREPORT.EQ.1)
C   X READ *, (FX(j), j=1, NBANDS)
C   IF(IREPORT.EQ.1.0E.7IREPORT.EQ.4) PRINT 128
128   FORMAT(4X,"ENTER THE RATIO OF THE BAND ERRORS (ONE NUMBER //"
X "(X, 'FOR EACH BAND). FOR EXAMPLE A 3 BAND FILTER//"
X "(X, 'MIGHT HAVE AN ERROR RATIO OF 1, 1.5, 5.')//")
C   IF(IREPORT.EQ.1.0E.7IREPORT.EQ.4)
C   X READ *, (ATR(j), j=1, NBANDS)
C   NFACT=NFILT/2+1
C
C   SET UP THE COarse GRID. THE NUMBER OF POINTS IN THE GRID
C   IS (FILTER LENGTH + 1) * GRID DENSITY / 2
C
C   GRID(1)=EDGE(1)
C   DELF=LGR1D*NFACT
C   DELF=1.5/DELF
C   J=1
C   L=1
C   LBAND=1
140   FUP=EDGE(L+1)
145   TEMP=GRID(J)
C
C   CALCULATE THE DESIRED MAGNITUDE RESPONSE AND THE WEIGHT
C   FUNCTION ON THE GRID
C
C   DES(J)=EFF(FX,LBAND)
C   WT(J)=WATE(NTX,LBAND)
C   J=J+1
C   GRID(J)=TEMP+DELF
C   IF(GRID(J).GT.FUP) GO TO 155
C   GO TO 145
150   GRID(J-1)=FUP
C   DES(J-1)=EFF(FX,LBAND)
C   WT(J-1)=WATE(NTX,LBAND)
C   LBAND=LBAND+1
C   L=L+2
C   IF(LBAND.GT.NBANDS) GO TO 154
C   GRID(J)=EDGE(L)
C   GO TO 145
154   NGRID=J-1

```

```

C INITIAL GUES FOR THE EXTRAYL SEARCH  TOL---EQUALY
C SPACED ALONG THE GRID
C
201 TEMP=FLOAT(NFCONS-1)/FLOAT(NFCONS)
DO 211 J=1, NFCONS
211 IEXT(J)=(J-1)*TOL*P+1
IEXT(NFCONS+1)=NG+1
NM1=NFCONS+1
N7=NFCONS+1
C CALL THE REM-7 EXCHANGE ALGORITHM TO DO THE APPROXIMATION
C PROBLEM
C
CALL REM7(EDGE,NBANDS)
C CALCULATE THE IMPULSE RESPONSE.
C
DO 301 J=1,NM1
305 H(J)=.1*ALPHA(17-J)
H(NFCONS)=.1*PHA(1)
C PROGRAM OUTPUT SECTION
C
341 PRINT 351 * WRITE(1,351)
351 FORMAT(1X,72(1H-)/15X,"ONE-DIMENSIONAL FIR PROTOTYPE"
X " FILE: DESIGN"/)
PRINT 378,NFILT WRITE(1,378)NFILT
373 FORMAT(27X,"FTLT FIR LENGTH = ",I3)
WRITE(1,374)
384 FORMAT(22X,"**** IMPULSE RESPONSE ****")
DO 381 J=1,NFCONS
K=NFILT+1-J
WRITE(1,382)(J-1,H(J),K-1)
381 CONTINUE
382 FORMAT(26X,"H(",I3,") = ",E15.8," = H(",I3,")")
DO 383 K=1,NP-1,1,4
KUP=K+3
IF(KUP.GT.NP) KUP=NP-BANDS
PRINT 384,(J,J=K,KUP)
WRITE(1,385)(J,J=K,KUP)
385 FORMAT(12+X,4(" 10",13,9X))
PRINT 386,(REFGE(2*j-1),J=K,KUP)
WRITE(1,387)(REFE(2*j-1),J=K,KUP)
391 FORMAT(2X,"LOWER BAND EDGE",FF15.9)
PRINT 392,(REFGE(2*j),J=K,KUP)
WRITE(1,393)(REFGE(2*j),J=K,KUP)
395 FORMAT(2X,"UPPER BAND EDGE",FF15.9)
PRINT 396,(WTX(J),J=K,KUP)
WRITE(1,400)(WTX(J),J=K,KUP)
403 FORMAT(2X,"DESIRED VALUE",2Y,5F15.9)
PRINT 410,(WTY(J),J=K,KUP)
WRITE(1,410)(WTY(J),J=K,KUP)
410 FORMAT(2X,"WTY ING",6X,5F15.9)
DO 420 J=K,KUP
421 DEVIAT(J)=DEV/WTY(J)
PRINT 425,(DEVIAT(J),J=K,KUP)
WRITE(1,425)(DEVIAT(J),J=K,KUP)
425 FORMAT(2Y,"DEVIATION",6X,5F15.9)
DO 430 J=K,KUP
IF(DEVIAT(J).EQ.0) GO TO 436
430 DEVIAT(J)=20.1*A.0G10(DEVIAT(J))
PRINT 435,(DEVIAT(J),J=K,KUP)
WRITE(1,435)(DEVIAT(J),J=K,KUP)
435 FORMAT(2X,"DEVIATION IN 10",5F15.9)
436 CONTINUE

```

```

453 CONTINUE
455 FORMAT(1/EX,"ENTER MAIN FREQUENCIES (NUMBERS MUST BE MULTIPLIED)"
      X " BY 2)"/(2X, F12.7)
      DO 451 J=1,N7
C
C SINCE THE MAIN PROGRAM USES RADIANS, HPIE, DEGREES ARE
C PARTIALLY CONVERTED TO RADIANS.
C
451 EXTN(J)=2.0*PI*RTD(IEXT(J))
      WRITE(1,451)(EXTN(J), J=1,N7)
      PRINT *," 3 42 TE(1,451)"
452 FOR MAT(72(1,452))
      PRINT *,""
480 FORMAT(1,EX,"DO YOU WISH TO RESIGN THE PROTOTYPE FILTER?",/)
      READ*,ACCEPT
      IF(ACCEPT.EQ.1) GO TO 249
      PRINT *,""
490 FORMAT(1,EX,"ENTER ONE OF THE FOLLOWING OPTION NUMBERS"
      X " 1) CHANGE VELOCITY/NG"
      X " 2) CHANGE ONLY THE FILTER ORDER"
      X " 3) CHANGE ONLY TRANSITION BAND EDGE FREQUENCIES"
      X " 4) CHANGE ONLY THE EDGE RATIO")
      READ*,REPORT
      GO TO 141
999 CONTINUE 2 END
C
C FUNCTION LFF(FX,LBAND)
C
C FUNCTION TO CALCULATE THE DESIRED MAGNITUDE RESPONSE
C AS A FUNCTION OF FREQUENCY.
C
      DIMENSION FY(5)
      SFF=FX(LBAND)
      RETURN
      END
C
C FUNCTION WATE(NTY,LTBAND)
C
C FUNCTION TO CALCULATE THE WEIGHT FUNCTION AS A FUNCTION
C OF FREQUENCY.
C
      DIMENSION WTX(5)
      WATE=WTX(LBAND)
      RETURN
      END
C
C SUBROUTINE RFME7(EDGE,NBANDS)
C
C THIS SUBROUTINE IMPLEMENTS THE RUMET EXCHANGE ALGORITHM
C FOR THE WEIGHTED OR BRYCHEV APPROXIMATION OF A CONTINUOUS
C FUNCTION WITH A SUM OF COSTINES. INPUTS TO THE SUBROUTINE
C ARE A DENSE GRID WHICH REPLACES THE FREQUENCY AXIS, THE DESIRED
C FUNCTION ON THIS GRID, THE WEIGHT FUNCTION ON THE GRID, THE
C NUMBER OF COEFFICIENTS, AND AN INITIAL GUESS OF THE EXTREMAL FREQUENCIES.
C THE PROGRAM MINIMIZES THE BRYCHEV ERROR BY DETERMINING THE BEST
C LOCATION OF THE EXTREMAL FREQUENCIES (POINTS OF MAXIMUM ERROR)
C AND THEN CALCULATES THE COEFFICIENTS OF THE BEST APPROXIMATION.
C
      COMMON PIZ,AD,DEV,X,Y,GRID,DES,WT,ALPHA,IEXT,NFCNS,NGRID
      DIMENSION EDGE(2)
      DIMENSION IEXT(56),AD(56),ALPHA(56),X(56),Y(56)
      DIMENSION DES(145),GRID(1045),WT(145)
      DIMENSION A(56),C(56),D(56)
      DOUBLE PRECISION PIZ,DNUM,OPEN,OTEMP,A,P,D
      DOUBLE PRECISION AD,DEV,X,Y

```

C THE PROGRAM ALLOWS A MAXIMUM NUMBER OF ITERATIONS OF 25
C

```
100 ITMAX=25
      DEVL=-1.0
      N7=NFCNS+1
      N77=NFCNS+2
      NIT=R=
110 CONTINUE
      IFXT(1,77)=NGRTD+1
      NIT=R=NIT+1
      IF(NIT.E.GT.ITMAX) GO TO 400
      DO 111 J=1,N7
      DTMP=GRJ1(IFXT(1))
      DTMP=00003(DTMP RIC)
111  X(J)=DTMP
      JFV=(NFCNS-1)/1041
      DO 112 J=1,N7
120  AD(J)=D(J,N7,177)
      DNU=.0
      DDF=.0
      K=1
      DO 130 J=1,N7
      L=IEXT(J)
      DTEM=AD(J)*DES(L)
      DNUM=DNUM+DTEM
      DTEM=K*AD(J)/NT(L)
      DDEN=DDEN+DTEM
130  K=-K
      DEV=DNUM/DDEN
      NU=1
      IF(DEV.GT.0.0) NU=-1
      DEV=-NU*DDEV
      K=N1
      DO 140 I=1,N7
      L=IEXT(I)
      DTEM=K*DDEV/NT(L)
      Y(I)=DES(L)+DTEM
140  K=-K
      IF(DEV.GE.DEVL) GO TO 150
      CALL DUCH
      GO TO 400
150  DEVL=DEV
      JCHANGE=
      K1=IEXT(1)
      KN7=IEXT(N7)
      KLOW=
      NUM=-1.0
      J=1
```

C SEARCH FOR THE EXTREMAL FREQUENCIES OF THE BEST
C APPROXIMATION

```
200 IF(J.EQ.N77) YN7=0000P
      IF(J.GE.N77) GO TO 300
      KUP=IFXT(J+1)
      L=IEXT(J)+1
      NUT=-NUT
      IF(J.EQ.2) Y1=0000P
      COMP=DEV
      IF(L.GE.KUP) GO TO 220
      ERM=GEE(L,N7)
      EPP=(ERM-DES(L))/WT(L)
      DTEMP=NUT*ERM-0000P
      IF(DTEMP.LE.0.1) GO TO 220
      COMP=NUT+EPP
210 L=L+1
```

```

        IF(L.GE.K1P) GO TO 215
        ERH=GEE(L,N7)
        ERF=(ERR-DES(L))*WT(L)
        DTEMP=NJT*ERR-0.01P
        IF(DTEMP.LE.0.1) GO TO 215
        COMP=NUT*ERR
        GO TO 215
215  TEXT(J)=L-1
        J=J+1
        KLOW=L-1
        JCHNGE=JCHNGE+1
        GO TO 215
221  L=L-1
225  L=L-1
        IF(L.LE.KLOW) GO TO 25
        ERR=GEE(L,N7)
        ERF=(ERR-DES(L))*WT(L)
        DTEMP=NJT*ERR-0.01P
        IF(DTEMP.GT.0.1) GO TO 225
        IF(JCHNGE.LF.1) GO TO 225
        GO TO 250
233  COMP=NUT*ERR
235  L=L-1
        IF(L.LE.KLOW) GO TO 241
        ERH=GEE(L,N7)
        ERF=(ERR-DES(L))*WT(L)
        DTEMP=NJT*ERR-0.01P
        IF(DTEMP.LE.0.1) GO TO 241
        COMP=NUT*ERR
        GO TO 235
241  KLOW=IEXT(J)
        IEXT(J)=J+1
        J=J+1
        JCHNGE=JCHNGE+1
        GO TO 241
251  L=IEXT(J)+1
        IF(JCHNGE.GT.1) GO TO 215
255  L=L+1
        IF(L.GE.K1P) GO TO 255
        ERH=GEE(L,N7)
        ERF=(ERR-DES(L))*WT(L)
        DTEMP=NJT*ERR-0.01P
        IF(DTEMP.LE.0.1) GO TO 255
        COMP=NUT*ERR
        GO TO 215
261  KLOW=IEXT(J)
        J=J+1
        GO TO 215
3.0  IF(J.GT.N7) GO TO 321
        IF(K1.GT.IEXT(1)) K1=IEXT(1)
        IF(KN7.LT.IEXT(N7)) KN7=IEXT(N7)
        NUT1=NUT
        NUT=-1.0
        L=
        K1P=K1
        COMP=YN7*(1.0E-11)
        LUCK=1
31)  L=L+1
        IF(L.GE.K1P) GO TO 315
        ERH=GEE(L,N7)
        ERF=(ERR-DES(L))*WT(L)
        DTEMP=NJT*ERR-0.01P
        IF(DTEMP.LE.0.1) GO TO 315
        COMP=NUT*ERR
        J=N7
        GO TO 215

```

```

317 LUCK=5
  GO TO 32E
321 IF(LUCK.GT.9) GO TO 321
  IF(COMP.C1,Y1) Y1=0012
  K1=TEXT(N77)
323 L=NGP1D+1
  K1CM=KN7
  NUT1=-NUT1
  COMP=Y1*(1.0E-11)
331 L=L-1
  TF(L,LE,KLOW) GO TO 331
  FT=AGEF(L,N7)
  KRF=(KRF-DEFL/L)/HT(L)
  DTTEMP=HT*ERP-COMP
  IF(DTTEMP.LE.0.1) GO TO 331
  J=N77
  COMP=1.0E-10
  LUCK=LUCK+1
  GO TO 22E
341 IF(LUCK.EQ.6) GO TO 377
  DO 341 J=1,NFCNS
345 IEXT(N77-J)=TEXT(N7-J)
  IEXT(1)=K1
  GO TO 111
351 KN=IEXT(N77)
  DO 351 J=1,NFCNS
354 IEXT(J)=TEXT(J+1)
  IEXT(N7)=KN
  GO TO 111
371 IF(JCHNG.EQ.0) GO TO 110

```

C C CALCULATION OF THE COEFFICIENTS OF THE BEST APPROXIMATION
C USING THE INVERSE DISCRETE FOURIER TRANSFORM

```

411 CONTINUE
  NM1=NFCNS-1
  FSH=1.1E-5
  STEMP=GRJ0(1)
  X(N77)=-2.1
  CN=2*NFCNS-1
  DELF=1. /CN
  L=1
  KKK=1
  IF((PGF(1).EQ.1).AND.EDGE(2*NFCNS).EQ.0.5) KKK=1
  IF(NFCNS.LE.3) K'K=L
  IF(KKK.EQ.1) GO TO 416
  DTTEMP=DCOS(PI2*G1D(1))
  DNUM=DCOS(PI2*GR1D(NGPIN))
  AA=2.1/(DTEMP+DNUM)/(DTEMP-DNUM)
  RR=-(DTEMP+DNUM)/(DTEMP-DNUM)
415 CONTINUE
  DO 431 J=1,NFCNS
  FT=(J-1)*DELF
  XT=DCOS(PI2*FT)
  IF(KKK.EQ.1) GO TO 411
  XT=(XT-RR)/AA
  FT=ACOS(XT)/PI2
411 XE=X(L)
  IF(XT.GT.XE) GO TO 421
  IF((XE-XT).LT.FSH) GO TO 415
  L=L+1
  GO TO 410
415 A(J)=Y(L)
  GO TO 42E
420 IF((XT-XE).LT.FSH) GO TO 415
  GRJD(1)=FT

```

```

      A(J)=GEE(1,N7)
425  CONTINUE
      IF(L.GT.1) L=L-1
430  CONTINUE
      GEE(1)=GEE*MP
      DO 11 J=1,NEONS
      GEE*MP=1.
      NM1=(J-1)*MP
      IF(NM1.LT.1) GO TO 11
      DO 11 K=1,NM1
      511  D1=MP*D1+D(K+1)*MP*GEE(DKUMA(K))
      505  D1=MP*2.5*D1*MP+D1
      511  ALPHA(J)=D1*MP
      DO 11 J=2,NEONS
      551  ALPHA(J)=2*ALPHA(J)/MP
      ALPHA(1)=ALPHA(1)/MP
      TF(KKK,20,1) GO TO 11
      F(1)=2.5*ALPHA(NEONS)*3.5+ALPHA(NM1)
      P(2)=2.5*ALPHA(NEONS)
      Q(1)=4*ALPHA(NEONS-2)-ALPHA(NEONS)
      DO 11 J=2,NM1
      IF(J.LT.1) GO TO 11
      AA=.5*AA
      BB=.5*BB
512  CONTINUE
      P(J+1)=.
      DO 12 K=1,J
      A(K)=F(K)
521  P(K)=2.5*P(K)*A(K)
      P(2)=P(2)+4*(1)*2.5*AA
      JM1=J-1
      DO 12 K=1,JM1
      525  P(K)=P(K)+Q(K)+4*(K+1)
      JP1=J+1
      DO 13 K=3,JP1
      531  F(K)=F(K)+AA*A(K-1)
      IF(J,FO,NM1) GO TO 13
      DO 13 K=1,J
      535  Q(K)=-A(K)
      Q(1)=Q(1)+ALPHA(NEONS-1-J)
541  CONTINUE
      DO 43 J=1,NEONS
      543  ALPHA(J)=P(J)
      545  CONTINUE
      IF(NEONS.GT.3) RETURN
      ALPHA(NEONS+1)=1.
      ALPHA(NEONS+2)=0.0
      RETURN
      END

```

DOUBLE PRECISION FUNCTION D(K,N,M)

FUNCTION TO CALCULATE THE LAGRANGE INTERPOLATION COEFFICIENTS FOR USE IN THE FUNCTION GEE.

```

COMMON PI2,AD,DEV,X,Y,GRD,DES,WT,ALPHA,ICXT,NEONS,GRID
DIMENSION ICXT(61),AD(66),ALPHA(65),X(66),Y(66)
DIMENSION DES(145),GRID(145),WT(145)
DOUBLE PRECISION AD,DEV,X,Y,Q,PI2
Q=1.0
Q=X(K)
DO 3 L=1,3
  DO 2 J=L,N,M
    IF(J-K11,2,1
1  D=2.0*(Q-Y(J)))

```


C TRANSFORMATION TO VTFLE TRANS(W_2, W_1).

C
COMMON /AAA/ H10CHER /PBY/ CURVFIT
COMMON /ONE/ IFTT /FIVE/H

C C
DIMENSION H10CHER(57),CURVFIT(18),H(66),H10(57),DAV(27,57)
DIMENSION H2DMAG(11,11)

C
C THE DIMENSIONS OF 18 ARE H, H10, H10CHER, AND DAV ARE ((NMAX-1)/2+1).
C H2DMAG IS DIMENSIONED (11,11) TO STORE THE 121 CALCULATED H(W_2, W_1)
C MAGNITUDE VALUES. CURVFIT IS DIMENSIONED (18). THE FIRST NINE LOCATIONS
C STORE THE CONSTANTS OF THE SECOND ORDER MOCELLAN TRANSFORMATION (IF THE FIFTH
C ORDER MOCELLAN TRANSFORMATION IS USED, FIVE OF THE NINE CONSTANTS
C WILL BE ZERO).

C
K = (Nfilt - 1)/2

K0 = K+1

C IN THIS SECTION H(J) IS CONVERTED TO H10(J).

C
H10(1) = H(K+1)

DO 1 J = 1, K

H10(J+1) = - $^2 H(K+1-J)$

10 CONTINUE

C IN THIS SECTION H10(J) IS CONVERTED TO H10CHER(J).

C FIRST ALL ENTRIES IN ARRAY DAV ARE SET TO ZERO.

C
DO 31 JA=1, K0

DO 21 JB=1, K0

DAV(JA,JB) = 0.0

20 CONTINUE

30 CONTINUE

C
C NEXT THE MAGNITUDES OF ALL THE NON-ZERO ENTRIES IN THE FIRST TWO ROWS
C ARE GENERATED.

C
DO 4 JA=1, K0, 2

DAV(1,JA) = 1.

40 CONTINUE

C
DO 5 JA=2, K0, 2

DAV(2,JA) = JA-1

50 CONTINUE

C
IF(K, EQ, 1) GO TO 120

C
C THEN ALL THE OTHER MAGNITUDES OF THE NON-ZERO ENTRIES ARE GENERATED
C USING A RECURSION FORMULA.

C
DO 7 JA=3, K0

DO 6 JB=JA, K0, 2

DAV(JA,JB) = DAV(JA,JB-2) + DAV(JA-1,JB-1)

60 CONTINUE

70 CONTINUE

C
C NOW ALL NON-ZERO ENTRIES ARE GIVEN THE PROPER SIGN.

C
M=3

DO 9 JA=1, K0

IF(M, GT, K0) GO TO 90

DO 8 JB=M, K0, 4

DAV(JA,JB) = - $^2 DAV(JA,JB)$

80 CONTINUE

M = M+1

90 CONTINUE

C C THEN THE NON-ZERO ELEMENTS IN EACH ROW ARE MULTIPLIED BY THE PROPER
C POWER OF TWO.

C POWER = 2.0

DO 110 JA=3, K0
DO 1 J3=J*, K*, 2
DAV(JA, J3) = DAV(JA, J3)*POWER

110 CONTINUE

POWER = 2.0* MFS

111 CONTINUE

C C FINALLY EACH VALUE OF H10CHER IS PRODUCED BY MULTIPLYING THE
C CORRESPONDING ROW OF DAV BY H10(J).

C 120 FC 1 2 3 J =1, K0

H10CHER(J) = 0.
DO 130 JP=J, M, 2
0 = DAV(J, JP) * H10(JP)
H10CHER(J) = H10CHER(J) + 0

130 CONTINUE

140 CONTINUE

C C IN THIS SECTION THE AMPLITUDE OF H(W2,W1) AT 121 POINTS IN THE FIRST
C QUADRANT OF THE W2-W1 PLANE IS CALCULATED.

PI = 3.141592653 98

DO 140 I=1, 1

W1 = PI * (I-1)/12.

DO 150 J=1, 11

W2 = PI * (J-1)/12.

H20MAG(I, J) = H10CHER(1)

DO 150 I=2, K0

H20MAG = H10CHER(M)*((CURVFIT(1) + CURVFIT(3)*COS(W2)
+ CURVFIT(5)*COS(2*W2) + CURVFIT(2)*COS(W1)
+ CURVFIT(4)*COS(W1)*COS(W2)
+ CURVFIT(6)*COS(W1)*COS(2*W2)
+ CURVFIT(7)*COS(2*W1)
+ CURVFIT(8)*COS(2*W1)*COS(W2)
+ CURVFIT(9)*COS(2*W1)*COS(2*W2))** (M-1))

H20MAG(I, J) = H20MAG(I, J) + H20MAG

150 CONTINUE

160 CONTINUE

170 CONTINUE

C C PROGRAM OUTPUT SECTION.

C PRINT 130

WRITE(1, 130)

180 FORMAT(//, 1H ,7 (1H*), /, 26X, 33HTWO-DIMENSIONAL FIR FILTER DESIGN)
PRINT 220, MFLT, NFLT

WRITE(1, 220) NFLT, MFLT

220 FORMAT(/, 21X, I3, H 3Y , I3, 21H SAMPLE POINTS FILTER)

PRINT 230

WRITE(1, 230)

230 FORMAT(//, 6X, 51HMAGNITUDE OF THE 2-D FREQUENCY RESPONSE IN THE FIR
1ST QUADRANT, /, 11X, 51H (THE FREQUENCY RESPONSE IS FOUR QUADRANT SYMM
2ETRIC), /, 6X, 51(1H-), //, 13H W1-AXIS (TIMES PI))

DO 240 IS=1, 11

IV=12-I

DC=(11-IS)/11.

PRINT 280, DC, (H20MAG(IV, IW), IW=1, 11)

WRITE(1, 280) DC, (H20MAG(IV, IW), IW=1, 11)

240 CONTINUE

250 FORMAT(4Y, 1H+, /, 4Y, 1H+, /, 4X, 1H+, /, F4.1, 1H+, 11F6.2)

C EACH TIME THE PROGRAM GOES THROUGH THIS MAIN LOOP, THE FOUR OR NINE
C TERMS MOCCLELLAN TRANSFORMATION IS MULTIPLIED TIMES THE CURRENT RUNNING
C PRODUCT. GATHERED IN LAST IN THE THROUGH THE MAIN LOOP. PLUS ONE OF
C THE K+1 SUMMATIONS TO KEEP PLACE FOR TIME. THE PROGRAM CYCLES THROUGH
C THE MAIN LOOP.

C

20 11 J=1, N
KC = K200 J+1
IF (MYOPT2 .EQ. 1) N = J
IF (MYOPT2 .EQ. 3) N = 2*J - 1

C

C IN THIS PAIR OF SUB-LOOPS, THE CURRENT PRODUCT GENERATED THE LAST
C TIME THROUGH THE INT. LOOP IS MOVED FROM PROD TO HD N.

C

DO 31 JA=1, N
DO 31 JB=1, N
WORK(JA,JB) = PROD(JA,JB)
CONTINUE
5
6
CONTINUE

C

C IN THE NEXT TWO PAIRS OF SUB-LOOPS, EACH TERM OF THE MOCCLELLAN
C TRANSFORMATION IS MULTIPLIED TIMES THE CURRENT PRODUCT STORED IN
C WORK. ALL OF THE PARTIAL RESULTS ARE STORED IN PROD.

C

DO 31 L=1, N
DO 31 M=1, N
PROD(L,M) = WORK(L,M) + MCTRAN(1,1)
CONTINUE
7
8
CONTINUE

C

DO 111 L=1, N
DO 37 M=1, N
PROD(L,M+1) = PROD(L,M+1) + WORK(L,M) * MCTRAN(1,2)
PROD(L+1,M) = PROD(L+1,M) + WORK(L,M) * MCTRAN(2,1)
PROD(L+1,M+1) = PROD(L+1,M+1) + WORK(L,M) * MCTRAN(2,2)

C

C IF MYOPT2 EQUALS FIVE, THERE ARE ONLY TERMS IN THE
C MOCCLELLAN TRANSFORMATION.

C

IF (MYOPT2 .EQ. 5) GO TO 100

C

PROD(L,M+2) = PROD(L,M+2) + WORK(L,M) * MCTRAN(1,3)
PROD(L+1,M+2) = PROD(L+1,M+2) + WORK(L,M) * MCTRAN(2,3)
PROD(L+2,M) = PROD(L+2,M) + WORK(L,M) * MCTRAN(3,1)
PROD(L+2,M+1) = PROD(L+2,M+1) + WORK(L,M) * MCTRAN(3,2)
PROD(L+2,M+2) = PROD(L+2,M+2) + WORK(L,M) * MCTRAN(3,3)

9
100
CONTINUE
100
CONTINUE

C

C AT THIS POINT, PROD CONTAINS THE COMPLETE PRODUCT FOR THE CURRENT
C TIME THROUGH THE MAIN LOOP. NOW A PARTIAL SUMMATION TAKES PLACE
C WITH THE RESULT OF ALL PARTIAL SUMMATIONS STORED IN H2DCHER.

C

110
DO 131 L=1, N
DO 121 M=1, N
H2DCHER(L,M) = H2DCHER(L,M) + H1DCHER(J+1) * PROD(L,M)
121
CONTINUE
131
CONTINUE
140
CONTINUE

END

C

C ****
C OVERLAY(MC_5,7)

C ****

PROGRAM PACKAGE

C THIS PROGRAM COMPUTES

C

C $(K2D)(K) (K2D)(K)$ L 4
C $MAG(H(N1, N2)) = S(N1) S(N2) H2D(L, 1)(COSW1)(COSW2)$ TO
C $L = 1 \quad 4 =$

C $(K2D)(K) (K2D)(K)$

C $MAG(H(N1, N2)) = S(N1) S(N2) H2D(L, 1)(COSW1)(COSW2)$. IF THE FIRST
C ORDER MOCELLIAN TRANSFORMATION IS USED, K2D=1. IF THE SECOND ORDER
C MOCELLIAN TRANSFORMATION IS USED, K2D=2.

C

C COMMON /M001/MYOPT2
C COMMON /M001/NFLT /M001/H2DCHEB

C

C DIMENSION CEC(57,57), H2DCHEB(57,57), H2D(57,57)

C

C THE DIMENSIONS OF THE ARRAYS CEC, H2DCHEB, AND H2D ARE (N=MAX-1)/2+1.
C IF MYOPT2 EQUALS NIN, THE SECOND ORDER MOCELLIAN TRANSFORMATION
C (K2D=2) WILL BE USED. IF MYOPT2 EQUALS FOUR, THEN THE
C FIRST ORDER MOCELLIAN TRANSFORMATION (K2D=1) WILL BE USED.

C

C IF (MYOPT2 .EQ. 1) K2D = 1
C IF (MYOPT2 .EQ. 2) K2D = 2
C K = ((NFLT-1)/2) * K2D
C KD = K+1

C

C THIS SECTION OF THE PROGRAM GENERATES THE ARRAY CEC.

C

C FIRST ALL ENTRIES IN ARRAY CEC ARE SET TO ZERO EXCEPT THE CENTER
C DIAGONAL ENTRIES WHICH ARE SET TO ONE.

C

C 1. DO 3 J1=1, KD
C 2. DO 2 J3=1, KD
C 3. CEC(JA,JB) = 0.0
C 4. CONTINUE
C 5. CEC(JA,JA) = 1.0
C 6. CONTINUE

C

C IF(K.EQ.1) GO TO 11

C

C THEN THE ENTRIES ON ALL NON-ZERO UPPER DIAGONALS ARE GENERATED
C USING TWO RECURSION FORMULAS.

C

C KE = K-1
C DO 6 J0=3, KD, 2
C CEC(1,JC) = CEC(2,JC-1)
C IF (JC+1 .GT. KD) GO TO 7
C CEC(2,JC+1) = CEC(3,JC) + 2*CEC(1,JC)

C

C N = JC+2
C IF (N .GT. KD) GO TO 7
C DO 5 J0=3, KE
C CEC(J0,N) = CEC(J0-1,N-1) + CEC(J0+1,N-1)
C N = N+1

C 5. CONTINUE
C KE = KE-2

C 6. CONTINUE

C

C THEN THE NON-ZERO ENTRIES IN EACH COLUMN ARE MULTIPLIED BY THE
C PROPER POWER OF TWO.

C

C 7. POWER = .0
C DO 9 JV=3, KD

LA = 1
IF (CEC(1,1) .EQ. 0) LA = 2
DO 8 JW=LA, JV, 2
CEC(JW,JV) = C(1,JV) * PCW, P

80 CONTINUE
PCWF = .5 * PCW, P
90 CONTINUE
115 CONTINUE

C
C IN THIS SECTION OF THE PROGRAM, H20CHB(L,M) IS PARTIALLY CONVERTED TO
C H20(L,M) BY GENERATING AN INTERMEDIATE ARRAY WHOSE VALUES ARE STORED
C IN H20(L,M). THE INTERMEDIATE ARRAY IS PRODUCED BY MULTIPLYING ROWS
C OF CEC BY ROWS OF H20CHB(L,M).

C
DO 30 JJ=1, K0
DO 40 I=1, K0
H20(JJ,I) = .0
DO 31 JP=J, K0, 2
P = CEC(J,JP) * H20CHB(JJ,JP)
H20(JJ,I) = H20(JJ,I) + P

31 CONTINUE
41 CONTINUE

51 CONTINUE

C
C HERE THE INTERMEDIATE ARRAY IS MOVED FROM H20(L,M) TO H20CHB(L,M).

C
DO 71 KA=1, K0
DO 81 KB=1, K0
H20CHB(KA,KB) = H20(KA,KB)

61 CONTINUE
71 CONTINUE

C
C IN THIS SECTION OF THE PROGRAM, THE FIRST INTERMEDIATE ARRAY (STORED
C IN H20CHB(L,M)) IS PARTIALLY CONVERTED TO H20(L,M) BY GENERATING A
C SECOND INTERMEDIATE ARRAY WHOSE VALUES ARE STORED IN H20(L,M). THE
C SECOND INTERMEDIATE ARRAY IS PRODUCED BY MULTIPLYING ROWS OF CEC
C BY COLUMNS OF H20CHB(L,M).

C
DO 101 I=1, K0
DO 31 JV=1, K0
H20(JV,I) = .0
DO 81 JP=J, K0, 2
P = CEC(J,JP) * H20CHB(JP,JV)
H20(JV,I) = H20(JV,I) + P

81 CONTINUE
91 CONTINUE

101 CONTINUE

C
C HERE THE SECOND INTERMEDIATE ARRAY IS MOVED FROM H20(L,M) TO
C H20CHB(L,M).

C
DO 120 LA=1, K0
DO 111 LB=1, K0
H20CHB(LA,LB) = H20(LA,LB)

111 CONTINUE

120 CONTINUE

C
C IN THIS SECTION OF THE PROGRAM THE SECOND INTERMEDIATE ARRAY (STORED
C IN H20CHB(L,M)) IS CONVERTED TO H20(L,M) BY MOVING THE ELEMENTS OF
C THE SECOND INTERMEDIATE ARRAY TO NEW LOCATIONS AND MULTIPLYING BY
C CONSTANTS.

C
H20(KD,KD) = H20CHB(1,1)
DO 130 LC=1, K
H20(KD,KD-LC) = .5 * H20CHB(1,LC+1)
H20(KD-LC,KD) = .5 * H20CHB(LC+1,1)

```

20 15  LF=1,K
H2D(KD-LD,KD-LD) = .2* H2D0H2D(LD+1,LD+1)
151  CONTINUE
C
C PROGRAM OUTPUT STATE 4.
C
9  P 141 17
171  FOR ALL (//,0X,-) THE TWO- DIMENSIONAL IMPULSE RESPONSE IS BEING
1  ,/,0X,-) WRITTEN TO YOUR HARDCOPY OUTPUT FILE (RESULT).)
M0111(1,1-7)
18. FOR MAT (//,48X,-) THE 2-DIMENSIONAL IMPULSE RESPONSE ,/,32X,
15: H2D0H2D IMPULSE RESPONSE SAMPLES ARE LOCATED IN THE FIRST QUADRANT
21,/,32X,F,(14-))
LS2=1
LS2=9
19: IF(LD,L-1,LE2) LE2=K
      WRITE(1,1,F) LD2-1,LS2,LS2+1,LS2+2,LS2+3,LS2+4,LS2+5,-S2+1,LS2+7
      DC 2 0 MD=1,ND
      WRITE(1,2,F) M-1, (H2D(MI,MC),MC=LS2,LE2)
20.  CONTINUE
192. FORMAT (//,11X,1H+,59X,1H,M/,11X,1H+,37X,3(1H-),/,1
1  2X,17H2D(L,1) +, FX,T3,F(1FX,I3),/,1X,128(1H+))
21. FORMAT (1X,44H2D(.,3,4H,M)+,C(F13.0))
      LS2 = LS2+3
      LE2 = LE2+3
      IF(LS2.E.KD) GO TO 192
      WRITE(1,22) 2*X-1,2*X+1,K+1,K+1,2*K,2*K,2*K,2*K
22: FORMAT (//,4X,1 "THESE ARE ,I3,7H TIMES ,I3,4FH 2-D IMPULSE RESPON
1SE SAMPLES. ONLY THE FIRST ,/,4X,I3,7H TIMES ,I3,56H SAMPLES HAVE
2RELY LISTED ABOVE. DUE TO SYMMETRIES, THE ,/,4X,64H REMAINING IMPUL
3SE RESPONSE SAMPLES CAN BE GENERATED BY USING THE ,/,4X,
434H LISTED SAMPLES AND THE RELATION: ,/,4X,17H H2D(L,M) = H2D(L,
5,I3,11H-M) = H2D(,I3,2H-L,,I3,10H-M) = H2D(,I3,9H-L,M).)
      END

```

CTC 190 //// END OF LIST ////

Vita

David Ciccolella was born on March 3, 1952 in Hempstead New York. He graduated from Bladensburg High School located in Bladensburg, Maryland in 1970. Then he attended the University of Maryland, College Park, from which he received the degree of Bachelor of Electrical Engineering in 1974. Upon graduation, he received a commission in the USAF. He was employed as a Data Analyst for Computer Sciences Corporation, Greenbelt Maryland until called to active duty in November 1974. He served as an electrical engineer in the 509th Civil Engineering Squadron, Pease AFB, New Hampshire until entering the School of Engineering, Air Force Institute of Technology, in June 1979.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/80D-13	2. GOVT ACCESSION NO. AD-A100 780	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DESIGN OF LINEAR PHASE, FINITE IMPULSE RESPONSE, TWO-DIMENSIONAL, DIGITAL FILTERS	5. TYPE OF REPORT & PERIOD COVERED MS Thesis	
7. AUTHOR(s) David Ciccolella Cart USAF	6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE December 1980	
	13. NUMBER OF PAGES 147	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified	
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES Approved for public release; DAW AFR 190-17 Signature: Lynch Fredric C. Lynch, Major, USAF Director of Public Affairs	16 JUN 1981	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Digital Filters Two-Dimensional McClellan Transformation Linear Phase Finite Impulse Response		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) An interactive computer program was developed that enables the user to design linear phase, finite impulse response, linear shift-invariant, two-dimensional digital filters. The program user can design lowpass, highpass, bandpass, bandstop, all-pass, and multi-band two-dimensional digital filters. The filters designed by using the program are nearly optimal in the Chebyshev sense and their magnitude versus frequency characteristics have quadrantal		

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

symmetry ($|H(w_2, w_1)| = |H(-w_2, w_1)| = |H(w_2, -w_1)| = |H(-w_2, -w_1)|$).

The technique implemented in the program consists of transforming a one-dimensional digital filter into a two-dimensional digital filter by a change of variables. This technique was first proposed by James H. McClellan and is called the McClellan transformation. The program user can elect to utilize either the first order or the second order McClellan transformation to design a two-dimensional digital filter.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)